Robust and Reliable Deep Learning by Synergizing with Pre-Trained Models

by

Rakshith Subramanyam

A Dissertation Presented in Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

Spring Berman, Co-Chair
Pavan Turaga, Co-Chair
Jayaraman J. Thiagarajan
Suren Jayasuriya
Yezhou Yang

ARIZONA STATE UNIVERSITY

August 2024

ABSTRACT

Over the past decade, the success of deep learning models has largely depended on the availability of extensive training data and the assumption that training and target data distributions are independent and identically distributed (i.i.d.). However, deviations from these conditions often reveal the models' brittleness, as they struggle with distribution shifts—discrepancies between training and testing datasets arising from various factors.

To address this challenge, this dissertation leverages Generative Adversarial Networks (GANs) to parameterize distribution shifts, using a novel single-shot target aware (SiSTA) adaptation technique. This approach updates a GAN with a target domain example to generate synthetic samples that facilitate the effective adaptation of predictive models to target conditions.

Beyond synthetic data generation, GANs are integral to digital imaging restoration tasks such as image denoising and super-resolution. However, they often perform poorly when the input data deviates from the training distribution. To address this, a technique called SPHInX (Style Projection Heads for Inverting X) is developed to enhance GAN inversion capabilities, thereby improving the model's ability to handle out-of-distribution images.

However, GANs struggle with semantic shifts caused by label shifts and class imbalances. Vision-Language Models (VLMs) are more effective in these scenarios. This dissertation introduces CREPE (CLIP Representation Enhanced Predicate Estimation), a framework that leverages VLMs to improve nuanced visual relationship prediction by better contextualizing the visual representations.

A key part of mitigating model failures involves understanding when and why these failures occur. This dissertation proposes a novel strategy for estimating failures

by parameterizing the decision rules learned by predictive models through VLMs. This approach refines the mechanism for failure estimation, allowing for more precise identification and correction of failures across various scenarios.

When there is a scarcity of data, the challenge becomes even more pronounced. In such cases, the problem is commonly modeled as a distribution of small tasks. This dissertation addresses this issue by exploring the use of knowledge graphs to dynamically modulate the weights of predictive models. This approach enables the models to adapt their decision rules effectively, enhancing flexibility and effectiveness in real-world applications.

Overall, this dissertation presents robust methodologies for understanding and mitigating adverse effects of distribution shifts on the performance of deep learning models, significantly advancing the adaptability and reliability of these models in dynamic environments. These contributions lay a foundation for future research into developing artificial intelligence systems that are capable of sustaining reliable performance across varying conditions.

ACKNOWLEDGMENTS

I would first like to express my deepest appreciation to Dr. Mark Naufel, who believed in me since my master's time and recruited me into the Luminosity Lab. His unwavering support and kindness have fostered an exceptionally positive work environment. I would like to thank Luminosity Lab for their generous funding and support throughout my Ph.D. journey.

I extend my sincere gratitude to Dr. Jayaraman Thiagarajan (Jay) for his dedication and immense impact on both my academic and personal growth. Since our collaboration began in the summer of 2020, Jay has been a pivotal figure in my research journey, mentoring me through my Ph.D. program, and facilitating valuable internships at Lawrence Livermore National Laboratories. His commitment extends beyond typical hours, generously sharing his time and expertise.

My heartfelt thanks also go to Dr. Spring Berman, who has been a guiding light since my master's program. Special thanks to Dr. Pavan Turaga for his exceptional mentorship, enriching my research skills and providing continuous support throughout my Ph.D. journey. I am grateful to my committee members, Dr. Suren Jayasuriya and Dr. Yezhou Yang, for their insightful feedback and contributions to my academic progress. I also appreciate the efforts of my academic advisors at ECEE, Sno Kleespies, and Jaya Krishnamurthy, for their indispensable assistance with administrative tasks.

I would like to thank Lawrence Livermore National Labs for supporting my research with unparalleled computing resources. My gratitude extends to Dr. Rushil Anirudh (Amazon) for his patient guidance and insights into new problem-solving techniques. Additionally, I acknowledge Stanford Research Institute and Dr. David Zhang for the learning opportunities during my internship in 2019.

I owe a special thanks to my high school teachers, Mr. Ravishankar Chatta

Subramaniam, Mrs. A.S. Visalakshi, and Mrs. S. Ruby, who sparked my curiosity and passion for learning. Their dedication deeply influenced my academic journey.

My heartfelt thanks go to my friends, who have played a crucial role in my journey. Especially my childhood companions, Jefrine and Mukundhan. We've shared a journey since toddlerhood, attending the same schools and learning together. Their brilliance always inspired me to aim higher. My gratitude extends to Kowshik Thopalli and Vivek Sivaraman. They have been more than friends, providing countless hours of discussion, support, and camaraderie that have enriched my personal and academic life. Kowshik has shared his wisdom in discussions about politics, religion, and philosophy, mentoring me in countless ways. I'd also like to recognize my friends Prajwal, Sai Kiran, Navya, Keerthi, Shiba, Karthik, Chase Adams, Tyler, Abhik, Nivedita Mahesh, Niveditha Muthukrishnan, Miruthula, Srivatsan, and countless more who each made significant contributions to my journey. Their support and friendship have been treasures along this challenging path. Finally, my partner Pratyusha deserves a special mention for her unwavering support throughout this journey. Her patience and understanding have provided a constant source of comfort and strength, and her encouragement has been invaluable. Through her compassion, she has been a constant source of warmth and positivity, transforming moments of stress into opportunities for growth.

I owe a heartfelt thanks to my family for their boundless love and support. My father, Subramanyam, dedicated countless hours driving me to school and coaching classes, despite his exhaustive night shifts, ensuring that I never missed an opportunity for learning. I also thank my mother Uma, who has been the heart of our home. She instilled in me a love for poetry and literature, and her delicious meals and nurturing environment have been the foundation of our family's happiness. Managing all this

along with her professional responsibilities, she has provided a stable and loving environment that supported my academic and personal growth. Their efforts have always provided me with comfort and a sense of stability. I am deeply indebted to the my grandmother Janaki and my grandfather late Sundaresan, whose inspiration to follow my dreams has been a guiding light in my pursuit of academic and personal excellence. My siblings, Anirudh and Raghavi, have enriched my childhood with joy and companionship. Growing up with them by my side has been one of my life's greatest blessings, making every challenge more manageable and every success more enjoyable.

TABLE OF CONTENTS

CHAPTER Page

LIST OF TABLES

Table Page

Table Page

LIST OF FIGURES

Figure Page

Chapter 1

INTRODUCTION

In recent years, the landscape of artificial intelligence (AI) has been profoundly transformed by the advent of deep learning models. These models have catalyzed significant advancements in numerous domains, including computer vision (Goodfellow et al. 2014; Karras, Laine, and Aila 2019; Karras et al. 2020; Karras et al. 2021), natural language processing (Vaswani et al. 2017; Devlin et al. 2019), and autonomous systems (Rao and Frtunikj 2018; Subramanyam 2018), leading to their wide and rapid adoption across a diverse array of sectors. In healthcare (Hosny et al. 2018; Young et al. 2020), deep learning is at the forefront of revolutionary changes, enhancing drug discovery through predictive analytics and personalized medicine approaches. Autonomous driving technologies have benefited from these algorithms, improving safety and efficiency in transportation. Other sectors experiencing transformative changes due to deep learning include financial services through fraud detection systems, retail with personalized customer experiences, and in robotics, where they are pivotal in developing more sophisticated and autonomous machines.

As deep learning models continue to drive significant technological breakthroughs, there is an ever-increasing demand for more sophisticated AI solutions that are capable of handling complex and nuanced tasks. This surge in demand has propelled the development of models that can not only push the boundaries in predictive modeling but can also enable unprecedented capabilities in reasoning and creative content generation.

To meet these demands, deep learning models have increasingly relied on large-scale

datasets and significant compute availability. The intuition is that such large volumes of training data will adequately represent the scenarios that the model will encounter upon deployment, and can thus lead to transferable data representations when leveraged using state-of-the-art (unsupervised) learning paradigms. Examples of this include large language models like GPT-4 (Brown et al. 2020) and LLaMA (Touvron et al. 2023), and image synthesis models such as StyleGANs (Karras, Laine, and Aila 2019) and Stable Diffusion (Rombach et al. 2022). Not surprisingly, these models have pushed the boundaries of AI applications, showcasing the importance of training models at *scale*.

However, in practical applications, we rarely have this luxury of exorbitant data and compute, and hence build predictive models with simplified assumptions. One such widely adopted, yet problematic, assumption is that the training and testing datasets are independently and identically distributed (i.i.d.) (Fei and Liu 2016). The violation of this assumption will lead to significant challenges in the real-world deployment of AI models. For example, when this assumption fails, models that exhibit excellent performance in closed-world settings (i.e., evaluate on train data) can falter when exposed to data from novel test environments. The discrepancy between training and testing data can arise due to a variety of factors, and can be characterized using the the mathematical framework of distribution shifts.

Understanding and addressing these distribution shifts is crucial as deep learning applications become increasingly integrated into critical infrastructures. Advanced methodologies such as domain adaptation, transfer learning, and continual learning frameworks are being developed to mitigate these issues, enhancing the reliability and utility of deep learning models across all sectors.

1.1 Motivation

Classification models serve as fundamental tools in supervised machine learning, mapping input features to categorical labels. These models are parameterized by a set of parameters θ, which are optimized during the training process to minimize a loss function L. This loss function quantifies the mismatch between the predicted labels $\hat{y}$ and the actual labels y. The process hinges on a training dataset $D_{\text{train}} = \{(x_i, y_i)|i = 1, \ldots, N\}$, where the objective is to identify the optimal parameter set θ^* that minimizes the expected loss across D_{train}. This is represented mathematically as:

$$\theta^* = \arg\min_{\theta} \frac{1}{N} \sum_{i=1}^{N} L(f(x_i; \theta), y_i).$$

This formulation, commonly know as empirical risk minimization (ERM) (Montanari and Saeed 2022), assumes that minimizing the average loss on the training set will generalize effectively to unseen data from the testing set. However, as discussed above, this assumption does not always hold true, particularly when the training and testing datasets originate from different distributions—a situation often encountered in practical applications. Such discrepancies between training and testing conditions highlight the limitations of conventional training paradigms and underscore the challenges in achieving robustness against distribution shifts.

Distribution shift manifests in various forms: **Covariate Shift** occurs when the distribution of input variables $P(x)$ differs between the training and testing datasets, while the conditional distribution $P(y|x)$ remains the same. **Label Shift** arises when the distribution of labels $P(y)$ changes between the datasets, while the conditional distribution of features given the labels $P(x|y)$ remains the same. **Sub-Population Shift** involves changes in the distribution within specific sub-groups of the data, which may not be reflected across the training and testing sets.

To address distribution shifts in machine learning, researchers have developed several strategies, including synthetic data augmentation and feature space alignment techniques. Synthetic data augmentation enhances model robustness by expanding the diversity of training data through the generation of artificial samples. This method utilizes various manipulations such as image rotation, scaling, and color adjustment to expose models to a broad range of variations, thereby promoting domain invariance.

Feature space alignment techniques, on the other hand, aim to reduce domain discrepancies by aligning the feature distributions of source and target domains. These methods typically involve adjusting the statistical properties of features, such as aligning the mean and covariance of distributions. This alignment can be achieved through sophisticated models that learn to map features from different domains into a common feature space, effectively making the model's predictions less sensitive to the differences between the training (source) and testing (target) data distributions.

Both approaches have limitations. Synthetic data augmentation requires careful design to ensure that the generated variations are relevant and do not introduce misleading biases. Feature space alignment, meanwhile, demands significant amounts of target domain data to effectively learn a representative joint feature space. Moreover, these techniques often necessitate complex model architectures or training procedures, increasing computational costs and requiring more sophisticated expertise in model development.

Hence, to effectively mitigate such shifts, it is crucial to parameterize the shift to understand how it occurs and develop strategies to address it. Within this context, image generative models like Generative Adversarial Networks (GANs), known for their capability to model large-scale image distributions, emerge as promising tools. Traditionally, GANs have been utilized for generating synthetic data to train classifiers.

This raises an intriguing question: Can GANs be effectively leveraged to handle and parameterize shifts in image distributions, enhancing their utility in dynamic environments?

Image generative models, such as GANs, play a pivotal role in addressing numerous inverse problems in digital imaging, including image denoising and super resolution, among other downstream tasks. However, these models often face significant challenges when there is a shift in the distribution of the input data. For instance, GANs that are trained on specific datasets like aligned human faces frequently struggle when confronted with slightly altered conditions, such as non-aligned or rotated faces. This underscore the need for developing more adaptable and robust solutions that can leverage pretrained GANs for digital imaging problems.

When the shift is semantic in nature, as observed with class imbalanced datasets or when attempting to generalize to unseen classes, traditional generative models struggle due to their limited semantic understanding. We need solutions that can comprehensively understand visual features and their associated semantics.

On the other hand, in few-shot classification problems, where each task involves classes represented by very few examples, the challenge becomes particularly pronounced. Here, the focus is on enabling models to learn generalizable rules that can be quickly adapted when exposed to a small set of new examples during inference.

Ultimately, to build robust solutions capable of mitigating shifts, it is crucial to understand where and why models fail, and comprehend the decision rules that the models learn from biased training data. This involves a deep dive into the mechanics of the task and the performance of the models in executing these tasks under varied conditions. Such an understanding is fundamental to developing more adaptive, resilient machine learning systems and underscores the necessity for accurate failure

estimation methods, which are essential for pinpointing weaknesses and areas for improvement in model performance.

1.2 Parameterizing distribution shift through image generative models

Deep learning models often degrade significantly under distribution shifts between training and test environments, as noted by Torralba et al (Torralba and Efros 2011). Adapting models to new domains typically requires target domain data, which isn't always feasible. Test-time adaptation has emerged as an alternative, enabling models trained on source data to adapt using only target data. However, the effectiveness of source-free adaptation (SFDA) methods like SHOT (Liang, Hu, and Feng 2020) and domain-exploiting approaches (Yang et al. 2021) can be limited by the availability of sufficient target data. Other techniques such as TENT (D. Wang et al. 2021) and MEMO (Zhang, Levine, and Finn 2021) often provide marginal improvements under complex shifts or scarce data (Thopalli, Turaga, and Thiagarajan 2022).

In Chapter 2, we address the challenge of adapting models in scenarios with minimal target data, particularly focusing on extreme single-shot cases. Common strategies involve synthetic augmentations (Gokhale et al. 2023) and generative augmentations using GANs, notably StyleGAN-v2 (Yue et al. 2022), which excel in generating high-quality, realistic images.

We introduce SiSTA (Single-Shot Target Adaptation) (Thopalli et al. 2023; Subramanyam, Thopalli, et al. 2023), a novel method leveraging target-aware generative augmentation to adapt classifiers in data-limited environments. SiSTA involves fine-tuning a pre-trained StyleGAN with a single target sample by inverting the image into the GAN's latent space to generate domain-representative images. The classifier is then

adapted using these synthesized samples, employing unique sampling strategies for better feature extraction. Extensively tested across benchmarks like CelebA, AFHQ, and DomainNet, SiSTA has consistently outperformed existing methods, sometimes surpassing them by over 15 percentage points and matching performance levels of an oracle scenario with abundant target data (Thopalli, Turaga, and Thiagarajan 2022). This validates our hypothesis that pre-trained generative models can be potent tools for enhancing classifier robustness in challenging adaptation scenarios.

1.3 Exploring GAN inversion in the face of distribution shift

In the digital imaging domain, inverse problems such as image denoising, super-resolution, compressed sensing, image editing, and inpainting are critical for enhancing visual data quality and usability. These problems are integral to diverse applications like medical imaging, satellite enhancement, and digital media restoration, highlighting their importance in both academic and practical settings.

Generative Adversarial Networks (GANs), especially advanced models like StyleGAN-v2, are pivotal in solving these inverse problems through GAN inversion, which maps an image back into the latent space of a GAN. However, challenges arise when inverting images that deviate from the training distribution, particularly with minor input alterations, highlighting the limitations of traditional GANs in handling out-of-distribution (OOD) images.

In such contexts, updating the GAN with only a single image becomes problematic, especially if the target image exhibits semantic shifts. This can complicate the adaptation of the GAN, as traditional training methods require multiple images to recalibrate effectively without overfitting to a single outlier.

The only viable solution, therefore, is to devise a method to improve the inversion process itself. This approach involves updating the latent prior to enhance the GAN's ability to accurately map these challenging OOD images back into its latent space, even when the input data significantly deviates from the model's training distribution.

In Chapter 3, we address the limitations of standard GAN inversion techniques, which often fail due to a lack of local smoothness at the inverted point in the latent space. To mitigate this issue, we introduce vicinal regularization in the latent space through a novel approach called SPHInX (Style Projection Heads for Inverting X) (Rakshith Subramanyam et al. 2022). This technique innovates by replacing the conventional mapping function between the latent and style latent spaces with a style projection head, which is proficient at generating meaningful representations for any input derived from the latent space. This enhancement is aimed at improving the effectiveness of GANs in solving complex inverse problems by ensuring more accurate and stable image reconstructions.

The utility of SPHInX is demonstrated across a broad spectrum of applications, showcasing its versatility and effectiveness in various challenging scenarios. It excels in reconstructing out-of-distribution (OOD) images with exceptional fidelity. Beyond simple reconstruction, SPHInX also enables semantically meaningful editing of OOD images. Additionally, it adeptly tackles intricate inverse problems including denoising, compressed recovery, and simultaneous inversion and attribute discovery. This marks a notable evolution in the use of generative adversarial networks for inverse imaging challenges, enhancing the precision and adaptability of GAN-based image processing techniques.

1.4 Employing Vision Language Models (VLMs) to contextualize semantic shifts in Visual Relationship prediction

When dealing with semantic shifts such as those found in long-tailed data distributions, traditional image generative models often fall short, particularly in tasks like classification where understanding the semantic content is crucial. In these cases, multimodal foundation models that can jointly process and integrate image features and semantic (textual) data offer a more effective solution. Vision-Language Models (VLMs) are exemplary in this context, as they excel in learning an aligned embedding space where image and text features converge, creating meaningful connections between visual and linguistic elements. In this context, CLIP (Radford et al. 2021a) has emerged as a powerful VLM that has been successfully employed in a variety of tasks. However, CLIP faces challenges in practical reasoning tasks, such as Visual Relationship Prediction (VRP), particularly due to its text embeddings' limitations in differentiating various predicates in subject-object pairings. To overcome these limitations, in Chapter 4, we propose CREPE (Subramanyam, Jayram, et al. 2023), which contextualizes the visual representations by obtaining a visually grounded text representation.

Upon evaluation using the Visual Genome (VG) benchmark—a challenging dataset for VRP—CREPE demonstrated remarkable results. It not only surpassed the baseline performances of models like UVTransE and VCTree but also achieved top-tier recall rates. Moreover, CREPE exhibited impressive generalization capabilities on the Unrel benchmark, effectively managing diverse and previously unseen predicate occurrences, even though it was not explicitly trained on such data. This underscores CREPE's

robustness and adaptability in learning and representing visual relationships, marking it as a significant advancement in the field of visual-language modeling.

1.5 Describing model failures using vision language models

The strategies discussed previously are aimed at mitigating the failures of classification models to ensure their robustness. Understanding where and how these failures manifest is crucial for developing effective solutions. Traditional approaches to detect potential model failures often rely on uncertainty estimation techniques. These methods utilize various metrics such as the confidence of model predictions, energy scores, or entropy measures. However, these approaches are inherently limited by their dependency on the model's internal representations and knowledge, which can ironically be the source of failure. Moreover, the often poorly calibrated nature of deep models complicates the reliance on model predictions for diagnosing failures. While such strategies can be effective in specific scenarios, they generally do not address the broad spectrum of failure modes that can emerge at test time and typically lack interpretability, which hinders our understanding and capability to rectify these issues.

In Chapter 5, we introduce a new holistic failure detection mechanism, *Prior Informed Model Evaluation* (*PRIME*), which addresses a diverse range of failure scenarios without solely relying on the model's predictions. PRIME incorporates knowledge from Vision-Language Models (VLMs) and features a novel training protocol that develops the Prior-Induced Model (PIM). Unlike traditional methods, PIM leverages early layer image features aligned with VLM embedding space using fine-grained, class-specific text attributes. This enhances decision-making reliability by computing similarity scores between image embeddings and text attributes to estimate

class-level predictions. Post-training, we use discrepancies between the standard model and PIM predictions as indicators of potential failures.

Additionally, the integration of VLM priors in PIM provides deeper insights into the original model's failure modes by analyzing attribute influences on predictions, enhancing interpretability. Extensive benchmarking against various failure scenarios like spurious correlations, image corruptions, and distribution shifts demonstrates that our method outperforms all considered baselines.

1.6 Learning knowledge graph structures to solve few-shot learning

In Chapters 6 and 7, we delve into advanced strategies for enhancing the performance of few-shot classification models, a domain where the challenge is to classify objects from very limited examples. Few-shot classification problems necessitate models that can quickly adapt to new tasks using only a handful of training samples.

While Large Language Models (LLMs) and Vision-Language Models (VLMs) have significantly advanced few-shot and even zero-shot classification capabilities, their effectiveness is often constrained to the distributions they were trained on. This limitation underscores the need for approaches that can construct and utilize relevant knowledge beyond the training distribution.

In such contexts, one promising approach is to develop separate, learnable components that accumulate and utilize generalizable knowledge. This knowledge can then be used to modulate the weights of a classification model tailored to a specific task. This concept is frequently leveraged in meta-learning problems, where the goal is to design models that can learn how to learn across tasks.

Meta-learning is exemplified by methods like Model-Agnostic Meta-Learning

(MAML), which aims to train a model on a variety of learning tasks, such that it can solve new learning tasks using only a small number of training samples. MAML focuses on finding a model initialization that can be effectively fine-tuned in a few gradient steps.

However, traditional meta-learning methods like MAML face challenges in generalizability across diverse tasks. To improve this, newer methodologies like Hierarchical Shot Meta-Learning (HSML) and Adaptive Risk Minimization Learning (ARML) have been introduced. These methods enhance adaptability by incorporating knowledge from prior experiences relevant to the target task, using an external knowledge structure to optimize task adaptation.

Building on these advancements, we introduce Contrastive Knowledge-Augmented Meta-Learning (CAML) (Subramanyam, Heimann, Jayram, Anirudh, and Thiagarajan 2023; Subramanyam et al. 2021), a novel approach designed to enhance the MAML framework. CAML enhances the generalization capabilities of meta-learners by using prototype graphs and an external meta-knowledge structure to create task-specific embeddings, which then modulate the base learner's parameters. Key innovations of CAML include Contrastive Knowledge Distillation, which infuses prior knowledge into the image embedding module to enrich task representations; a Parameter-free Task Encoding Scheme that uses average pooling to generate robust task representations, simplifying the model architecture; and a Moving Average-based Update Strategy that continually updates the knowledge structure, encoding a broader range of experiences and boosting the model's adaptability. These enhancements collectively refine CAML's approach to meta-learning, providing a sophisticated method to tailor model behavior to specific task requirements and past learning experiences, thereby significantly improving few-shot classification performance.

Chapter 2

TARGET-AWARE GENERATIVE AUGMENTATIONS FOR SINGLE-SHOT ADAPTATION

Deep models tend to suffer a significant drop in their performance when there is a shift between train and test distributions (Torralba and Efros 2011). A natural solution to improve generalization under such domain shifts is to adapt models using data from the target domain of interest. However, it is infeasible to obtain data from every possible target during source model training itself. Test-time adaptation has emerged as an alternate solution, where a source-trained model is adapted solely using target data without accessing the source data. However, the success of these source-free adaptation (SFDA) methods hinges on sufficient target data availability (Liang, Hu, and Feng 2020; Yang et al. 2021). While there exist online adaptation methods such as TENT (D. Wang et al. 2021) and MEMO (Zhang, Levine, and Finn 2021), they are are found to be ineffective under complex distribution shifts and when target data is limited, often producing on par or only marginally better results than non-adaptation performance (Thopalli, Turaga, and Thiagarajan 2022).

In this chapter, we investigate a practical, yet challenging, scenario where the goal is to adapt models under unknown distribution shifts with minimal target data. Specifically, we focus on the extreme case where only single-shot example is available. In such data scarce settings, it is common to leverage synthetic augmentations; examples range from image manipulations to adversarial corruptions (Gokhale et al. 2023). Despite their wide-spread adoption, the best augmentation strategy can vary for different shifts, and more importantly, their utility diminishes in the single-shot

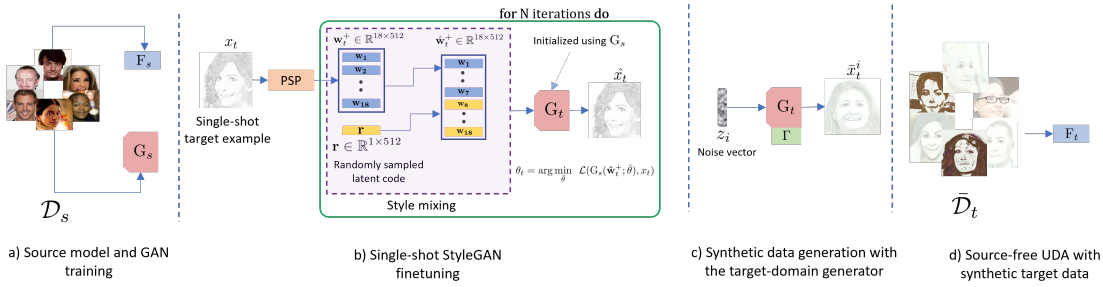


Figure 1. `SiSTA`: Assuming access to both the classifier and a StyleGAN from the source domain, we first adapt the generator to the target domain using a single-shot example. Next, we employ the proposed activation pruning strategies to construct the synthetic target dataset $\bar{\mathcal{D}}_t$. Finally, this dataset is used with any SFDA technique for model adaptation.

case. Another popular approach is to use generative augmentations (Yue et al. 2022), where data variants are synthesized through generative models. Despite being more expressive than generic augmentations, they require comparatively larger datasets for effective training.

We propose `SiSTA`, a new target-aware generative augmentation technique for SFDA with single-shot target data (see Figure 1). At its core, `SiSTA` relaxes the assumption of requiring source data, and instead assumes access to a source-trained generative model. We motivate and justify this assumption using a practical vendor-client implementation in Section 2.2. In this study, we consider StyelGAN as the choice for generative modeling, motivated by their flexibility in disentangling content and style. Our proposed algorithm has two steps, namely `SiSTA-G` and `SiSTA-S`, to fine-tune a source-trained StyleGAN with the target data, and to synthesize diverse augmentations respectively.

Our contributions can be summarized as follows:

1. We propose a new target-aware, generative augmentation technique for single-shot adaptation;

2. We introduce two novel sampling strategies based on activation pruning, *prune-zero* and *prune-rewind*, to support domain-invariant feature learning;

3. Using a popular SFDA approach, NRC (Yang et al. 2021), on augmentations from `SiSTA`, we show significant gains in generalization over SoTA online adaptation;

4. By benchmarking on multiple datasets (CelebA, AFHQ, CIFAR-10, DomainNet) and a wide variety of domain shifts (style variations, natural image corruptions), we establish `SiSTA` as a SoTA method for $1-$shot adaptation;

5. We show the efficacy of `SiSTA` in multi-class classification using both class-conditional GANs as well as multiple class-specific GANs.

2.1 Background

Source free domain Adaptation: In the standard setting of SFDA we only have access to the pre-trained source classifier $\mathrm{F}_s : \mathrm{x} \rightarrow \mathrm{y}$ but not to the source dataset $\mathcal{D}_s = \{(\mathrm{x}_s^i, \mathrm{y}_s^i)\}$. Here, $\mathrm{x}_s^i \in \mathcal{X}_s$ and $\mathrm{y}_s^i \in \mathcal{Y}$ denote the i^{th} image and its corresponding label from the source domain $\mathcal{X}_s$. Subsequently, the model needs to be adapted to a target domain $\mathcal{X}_t$ using unlabeled examples $\mathcal{D}_t = \{(\mathrm{x}_t^j\}$, where $\mathrm{x}_t^j \in \mathcal{X}_t$. Note, the set of classes $\mathcal{Y}$ is pre-specified and remains the same across all domains.

A number of approaches to SFDA have been proposed in the literature and can be categorized into two groups: methods which perform adaptation by fine-tuning the source classifier alone, and those that update the feature extractor as well for promoting domain invariance. In the former category, adaptation is typically achieved through unsupervised/self-supervised learning objectives; examples include rotation prediction (Sun et al. 2020), self-supervised knowledge distillation (Liu and Yuan 2022), contrastive learning (Huang et al. 2021) and batch normalization statistics

matching (D. Wang et al. 2021; Ishii and Sugiyama 2021). The second category includes state-of-the-art approaches such as SHOT (Liang, Hu, and Feng 2020), NRC (Yang et al. 2021) and N2DCX (S. Tang et al. 2021), which utilize pseudo-labeling based optimization, and often require sufficient amount of data to update the entire feature extractor meaningfully.

While SHOT is known to be effective under challenging shifts, it relies on global clustering to obtain pseudo-labels for the target data, and in practice, can fail in some cases due to the prediction diversity among samples within a cluster. The more recent NRC (Yang et al. 2021) alleviates this by exploiting the neighborhood structure through the introduction of affinity values that reflect the degree of connectedness between each data point and its neighbors. This inherently encourages prediction consistency between each samples and its most relevant neighbors. Formally, the optimization of NRC involves the following objective:

$$\mathcal{L}_{\text{NRC}} = \mathcal{L}_{\text{neigh}} + \mathcal{L}_{\text{self}} + \mathcal{L}_{\text{exp}} + \mathcal{L}_{\text{div}} \tag{2.1}$$

where $\mathcal{L}_{\text{neigh}}$ enforces prediction consistency of a sample with respect to its neighbors, while $\mathcal{L}_{\text{self}}$ attempts to reduce the effect of noisy neighbors and $\mathcal{L}_{\text{exp}}$ considers expanded neigbhorhood structure. Finally, $\mathcal{L}_{\text{div}}$ is the widely adopted diversity maximization term implemented as the KL divergence between the distribution of predictions in a batch to a uniform distribution. While `SiSTA` can admit any SFDA technique, we find NRC to be an appropriate choice, since it updates the feature extractor and utilizes the local semantic context to improve performance. This is particularly important in the context of our rich synthetic augmentations, which exhibit a high degree of diversity.

Generative Augmentations: It is well known that the performance of SFDA methods suffers when the target dataset is sparse. To mitigate this, synthetic aug-

mentations are often leveraged. While it has been found that data augmentation can improve both in-distribution and out-of-distribution (OOD) accuracies (Steiner et al. 2021; Hendrycks et al. 2021), their use in SFDA is more recent. Existing augmentations can be broadly viewed in two categories - (i) pixel/geometric corruptions, and (ii) generative augmentations. The former category includes strategies such as CutMix (Yun et al. 2019a), Cutout (DeVries and Taylor 2017), Augmix (Hendrycks et al. 2019), RandConv (Z. Xu et al. 2021), mixup (H. Zhang et al. 2018) and AutoAugment (Cubuk et al. 2019). These domain-agnostic methods are known to be insufficient to achieve OOD generalization, especially under complex domain shifts. To circumvent this, generative augmentations based on GANs or Variational Autoencoders (VAEs) have emerged. These methods involve training a generative model to synthesize new samples (Yue et al. 2022). These augmentations have been used in various tasks such as image-to-image translation and improving generalization under shifts. For example, methods such as MBDG (Robey, Pappas, and Hassani 2021), CyCADA (Hoffman et al. 2018), 3C-GAN (Rahman, Rahman, and Mahdy 2021) and GenToAdapt (Sankaranarayanan et al. 2018) have leveraged generative augmentations to better adapt to unlabeled target domains. However, by design, these methods require large amounts of data from both source and target domains. In contrast, `SiSTA` focuses on obtaining target-aware generative augmentations by fine-tuning source-trained generative models using only a single-shot target sample.

StyleGAN-v2 Architecture: While significant progress has been made in generative AI, including StyleGANs and denoising diffusion models (Saharia et al. 2022), we utilize StyleGAN-V2 as the base generative model in our work. This choice is motivated by the flexibility that StyleGANs offer in producing images of different styles, which can be attributed to the inherent disentanglement of style and semantic content in their

latent space. Existing approaches works (Wu, Lischinski, and Shechtman 2021; Wu et al. 2021) have studied this disentanglement property and uncovered the StyleGAN's ability to manipulate the style of an image projected onto the latent space by replacing the latent codes corresponding to only style. Another recent study (Chong and Forsyth 2021) reported that by leveraging such manipulations, one can perform style transfer with a limited number of paired examples. Interestingly, it has also been recently found (Wu et al. 2021) that, even after transferring a GAN to a different data distribution (faces to cartoons), the latent space of the adapted GAN is point-wise aligned with the source StyleGAN. We take inspiration from these works to develop our single-shot GAN fine-tuning protocol as well as our novel sampling strategies to enable domain-invariant feature learning.

2.2 Proposed Approach

In this section, we introduce `SiSTA`, a new target-aware, generative augmentation strategy with the goal of improving domain adaptation of pre-trained classifiers using single-shot target data. While SFDA methods are known to be effective under a variety of distribution shifts, their performance hinges on the availability of a sufficient amount of target data. In this work, we propose to relax SFDA's assumption on source data access by requiring a source-trained generative model (StyleGANs in our study) to synthesize augmentations in the target domain, in order to enable effective adaptation even under limited data. In particular, we consider the extreme, yet practical setting where only $1-$shot target data is available.

Figure 2 illustrates an implementation of such a setup where the source dataset, classifier, and the pre-trained generator are available only on the *vendor* side. A *client*

Algorithm 1 `SiSTA-G`

Input: Target sample x_t, Number of training iterations M, Source generator G_s, Inversion module $\mathcal{E}$, Set of style layers $\mathcal{L}_{\text{st}}$.
Output: Fine-tuned generator G_t.
Invert the target sample to obtain $\mathbf{w}_t^+ = \mathcal{E}(\mathrm{x}_t)$;
for m **in** 1 **to** M **do**

- Generate random style latent $\mathbf{r}^+$;
- Perform style-mixing, i.e., replace style layers $\mathcal{L}_{\text{st}}$ of $\mathbf{w}_t^+$ with $\mathbf{r}^+$;
- Generate image $\hat{\mathrm{x}}_t = \mathrm{G}_s(\hat{\mathbf{w}}_t^+)$;
- Update parameters Θ_t using equation (equation number or reference);

return: G_t with parameters Θ_t.

that wants to adapt the classifier to a novel domain submits the one-shot target data and receives both the source classifier as well as the synthetic generative augmentations. Finally, the *client* executes any SFDA approach to update the classifier using only the unlabeled synthetic data. This implementation eliminates the need for the *vendor* to share their generative model, while also minimizing the amount of *client* data that gets shared.

As described earlier, `SiSTA` is comprised of two key steps that are carried out on the *vendor* side: (i) `SiSTA-G`: Fine-tune a pre-trained StyleGAN generator G_s using single-shot target data $\{\mathrm{x}_t\}$ under unknown distribution shifts.; and (ii) `SiSTA-S`: Synthesize diverse samples $\mathcal{D}_t = \{\bar{\mathrm{x}}_t^j\}$ using the fine-tuned generator G_t to support effective classifier adaptation to the target domain. Finally, we leverage the recently proposed NRC method to perform *client*-side adaptation. Now, we describe these steps in detail.

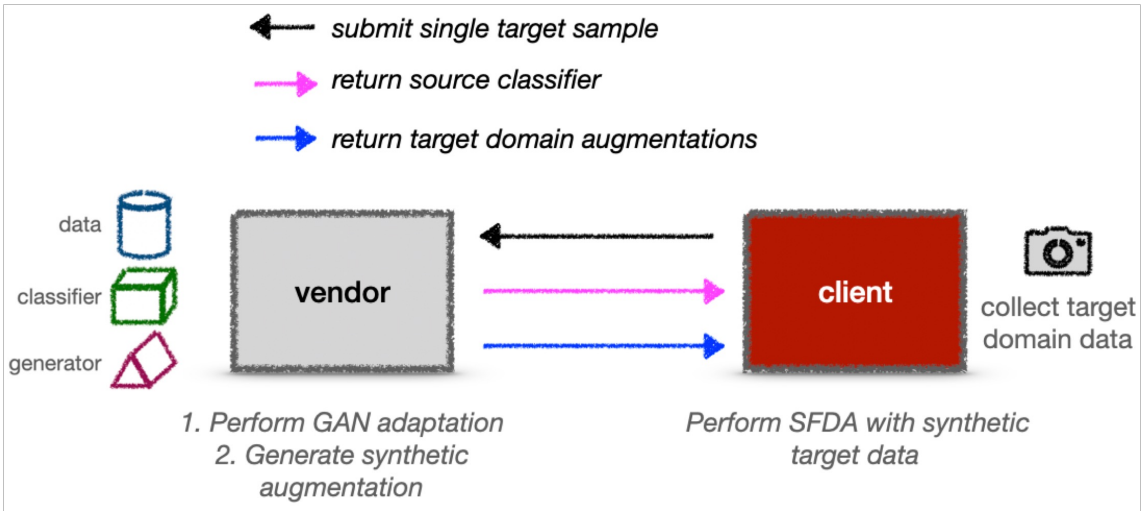


Figure 2. **A high-level illustration of our adaptation approach** `SiSTA`, which is carried out on the *vendor* side that stores the source classifier and a generative model. Designed to support single-shot adaptation, `SiSTA` returns target-aware synthetic augmentations. Finally, the *vendor* executes any SFDA technique to update the source classifier using the synthesized augmentations.

2.2.1 `SiSTA-G`: Single-Shot StyleGAN Fine-Tuning

Our goal in this step is to fine-tune G_s using only the single-shot example x_t from the target domain to produce an updated generator G_t. To this end, the proposed approach first inverts x_t onto the style-space of G_s. In practice, this can be done using one of the following strategies: (i) a pre-trained encoder such as Pixel2Style2Pixel (Richardson et al. 2021) or E4E (Tov et al. 2021), which maps a given image into the style code $\mathbf{w}_t^+ \in \mathbb{R}^{L\times 512}$. This latent code corresponds to L intermediate layers of a StyleGAN model (e.g., $L = 18$ in StyleGAN-v2); (ii) any standard GAN inversion technique to infer an approximate solution in the style space (Xia et al. 2022); (iii) text-guided inversion such as StyleClip (Patashnik et al. 2021) if the label is available for the single-shot target image. Though conventional GAN inversion is known to be expensive, it will not be a significant bottleneck with only a single image.

Without loss of generality, the target domain is expected to contain distribution shifts w.r.t. the source domain, and hence the inverted solution in the style-space is more likely to resemble the source domain. For example, inverting a cartoon into the style-space of a GAN trained on real face images will produce a semantically similar image from the face manifold. Recent evidence (Rakshith Subramanyam et al. 2022) suggests that one can accurately recover an OOD image using an additional vicinal regularization to the inversion process. However, in our case, we do not want an accurate reconstruction, but rather refine the generator G_s to emulate the characteristics of a target domain.

To this end, we utilize the following loss function defined on the activations from the source-domain discriminator H_s:

$$\Theta_t = \arg\min_{\bar{\Theta}} \quad \sum_{\ell} \|\mathrm{H}_s^{\ell}(\mathrm{G}_s(\mathbf{w}_t^{+}; \bar{\Theta})) - \mathrm{H}_s^{\ell}(\mathrm{x}_t)\|_1, \tag{2.2}$$

where $\mathbf{w}_t^{+}$ is the style-space latent code obtained via GAN inversion, Θ_t refers to the parameters of the updated generator G_t and H_s^{ℓ} denotes the activations from layer ℓ of the discriminator H_s. Intuitively, this objective minimizes the discrepancy between the target image and the reconstruction from the updated generator. Note that, the parameters of the discriminator are not updated during this optimization. While any pre-trained feature extractor can be used for this optimization, the source discriminator provides meaningful gradients by comparing both the content and style aspects of the target image. Upon training, we expect the generator G_t to produce images resembling the target domain for any random latent code in the style-space.

An inherent issue with our objective is that, this optimization can be highly unstable when using a single x_t. To circumvent this, we leverage multiple, style-manipulated versions of x_t through a style-mixing protocol. More specifically, we first generate a random code $\mathbf{r}^{+}$ in the style-space (using the mapping network in

StyleGAN). Next, we perform mixing by replacing the latent codes from a pre-specified subset of layers $\mathcal{L}_{\text{st}}$ in $\mathbf{w}_t^+$ using the corresponding codes from $\mathbf{r}^+$. In effect, this produces a modified image that contains the content from $\mathbf{w}_t^+$ and the style from $\mathbf{r}^+$. We denote this style-manipulated latent using the notation $\hat{\mathbf{w}}_t^+$. In each iteration of our optimization, a different style-mixed latent code $\hat{\mathbf{w}}_t^+$ is generated to compute the loss in (2.2). Algorithm 1 summarizes the steps of `SiSTA-G`.

Choosing layers for style-mixing. We choose $\mathcal{L}_{\text{st}}$ by exploiting the inherent style and content disentanglement in StyleGANs. Priors works (Wu, Lischinski, and Shechtman 2021; Kafri et al. 2021; Karras et al. 2020) have established that the initial layers typically encode the semantic content, while the later layers capture the style characteristics. Since the exact subset of layers that correspond to style vary as the image resolution changes, following standard practice, we used $\mathcal{L}_{\text{st}} = 8 - 18$ when G_s produces images of size 1024×1024 and $\mathcal{L}_{\text{st}} = 3 - 8$ for images of size 32×32 (CIFAR-10).

2.2.2 `SiSTA-S`: Target-aware Augmentation Synthesis

Once we obtain the target domain-adapted StyleGAN generator G_t, we next synthesize augmentations by sampling in its latent space. Despite the efficacy of such an approach, the inherent discrepancy between the true target distribution $P_t(\text{x})$ and the approximate $Q_t(\text{x})$ (synthetic data) can limit generalization. Existing works (Kundu et al. 2020) have found that constructing generic representations (using standard augmentations) is useful for test-time adaptation any domain. However, in contrast, our goal is to produce augmentations specific only to a given target domain, thus enabling effective generalization even with single-shot data.

Algorithm 2 GAN Pruning and Style Transfer

Require: Target GAN $\mathrm{G}_t(.;\Theta_t)$, Source GAN $\mathrm{G}_s(.;\Theta_s)$, Pruning strategy Γ, Pruning ratio p, Set of style layers $\mathcal{L}_{\text{st}}$
Ensure: Sampled image $\bar{\mathrm{x}}_t$
Draw a random latent code $\mathbf{w}^+$ from $\mathrm{G}_t(.;\Theta_t)$
for ℓ in $\mathcal{L}_{\text{st}}$ **do**

- $\beta \sim \text{Uniform}(0,1)$ ▷ Draw a Bernoulli random variable
- **if** $\beta < 0.5$ **then**
  - Obtain layer ℓ activations h_t^ℓ from $\mathrm{G}_t(\mathbf{w}^+)$
  - **for** v in 1 to V^ℓ **do**
    - $\tau_p = p$-th percentile of $\mathrm{h}_t^\ell[:,:,v]$
    - **if** Γ == *prune-zero* **then**
      - $\mathrm{h}_t^\ell[i,j,v] = 0$ **if** $\mathrm{h}_t^\ell[i,j,v] < \tau_p$, $\forall\ i,j$
    - **else**
      - Obtain activations h_s^ℓ from $\mathrm{G}_s(\mathbf{w}^+)$
      - $\mathrm{h}_t^\ell[i,j,v] = \mathrm{h}_s^\ell[i,j,v]$ **if** $\mathrm{h}_t^\ell[i,j,v] < \tau_p$, $\forall\ i,j$

return Image $\bar{\mathrm{x}}_t = \mathrm{G}_t(\mathbf{w}^+;\Gamma)$

To this end, we propose two novel strategies that perturb the latent representations from different layers of G_t to realize a more diverse set of style variations. Both our sampling strategies are based on activation pruning, *i.e.*, identifying the activations in each style layer that are lower than the p^{th} percentile value of that layer, and replacing them with (i) zero (referred to as *prune-zero*); or (ii) activations from the corresponding layer of the source GAN G_s (*prune-rewind*). The former strategy aims at creating a generic representation by systematically eliminating style information in the image. On the other hand, the latter attempts to create a smooth interpolation between the source and target domains by mixing the activations from the two generators. Note, we perform pruning only in the style layers, so that the semantic content of a sample is not changed. Note, we use the same set of style layers selected for performing `SiSTA-G`. Algorithm 2 lists the activation pruning step.

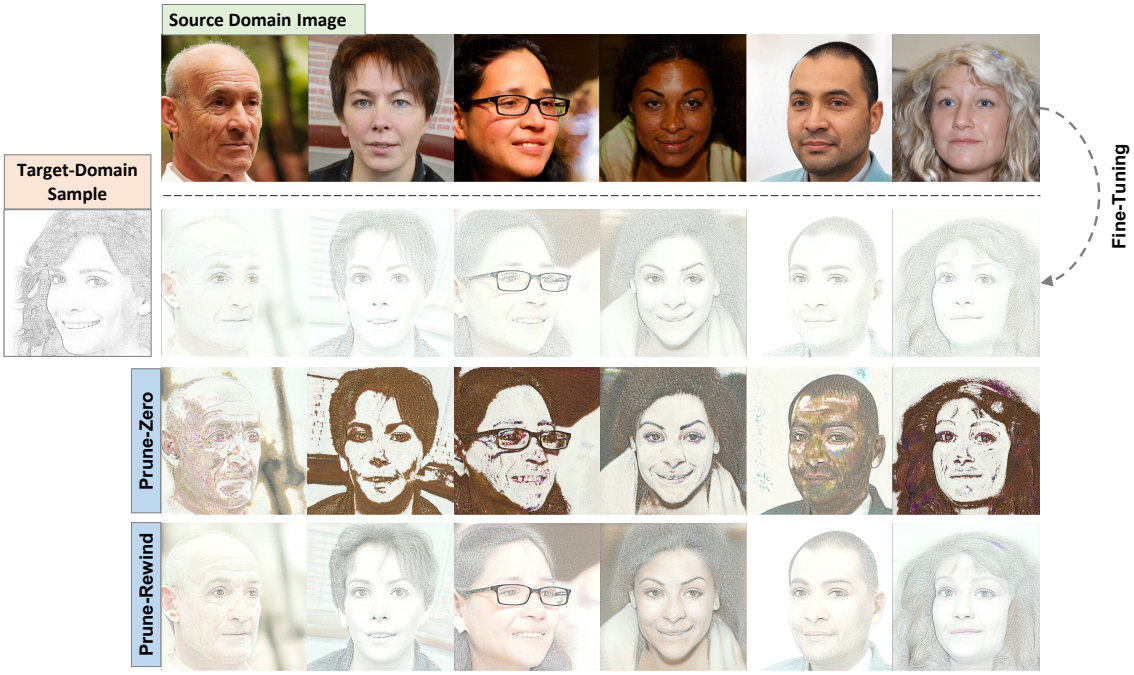


Figure 3. **Synthetic data generated using our proposed approach**. In each case, we show the source domain image and the corresponding reconstructions from the target StyleGAN sampling (base), prune-zero and prune-rewind strategies.

2.2.3 `SiSTA-mcG`: Extending to class-conditional GANs

When dealing with multi-class problems, it is typical to construct class-conditional GANs, $\mathrm{G}_s(.;\mathrm{c})$, to effectively model the different marginal distributions. In such settings, images from different classes get mapped to disparate sub-manifolds in the StyleGAN latent space. Assuming there are K different classes in $\mathcal{Y}$, we can directly apply `SiSTA-G` using 1-shot examples from each of the classes. The only difference occurs in the GAN inversion step, wherein we need to identify the conditioning variable c along with the latent code $\mathbf{w}_t^+$. Note, if the labels are available, one can estimate only $\mathbf{w}_t^+$. Finally, the algorithm 1 is repeated with K target images. We refer to this protocol as `SiSTA-mcG` (multi-class generation).

However, when we perform `SiSTA-mcG` using only a subset of the classes (say only

one out K), there is a risk of not incorporating target-domain characteristics into the images synthesized for all realizations from the latent space. However, as we will show in the results (Figure 5a), even using an example from a single class still leads to significantly improved generalization. We hypothesize that this behavior is due to the fact that the synthesized augmentations (random samples from G_t) arise from both $\mathcal{X}_s$ and $\mathcal{X}_t$, thus emulating an implicit mixing between the two data manifolds.

2.3 Experiments

We perform an extensive evaluation of `SiSTA` using a suite of classification tasks with multiple benchmark datasets, different StyleGAN architectures and more importantly, a variety of challenging distribution shifts. In all our experiments, we use single-shot target data and utilize publicly available, pre-trained StyleGAN weights.

2.3.1 Experimental Setup

Datasets: For our empirical study, we consider the following four datasets: (i) CelebA-HQ (Karras et al. 2017b) is a high-quality (1024×1024 resolution) large-scale face attribute dataset with 30K images. We split this into a source dataset of 18K images and the remaining was used to design the target domains. We perform attribute detection experiments on a subset of 19 attributes, i.e., each attribute is posed as its own binary classification task; (ii) AFHQ (Choi et al. 2020) is a dataset of animal faces consisting of 15,000 images at 512×512 resolutions with three classes, namely cat, dog and wildlife, each containing 5000 images. For each class, 500 images were used to create the target domains, and the remaining was used as the source data; (iii) CIFAR-

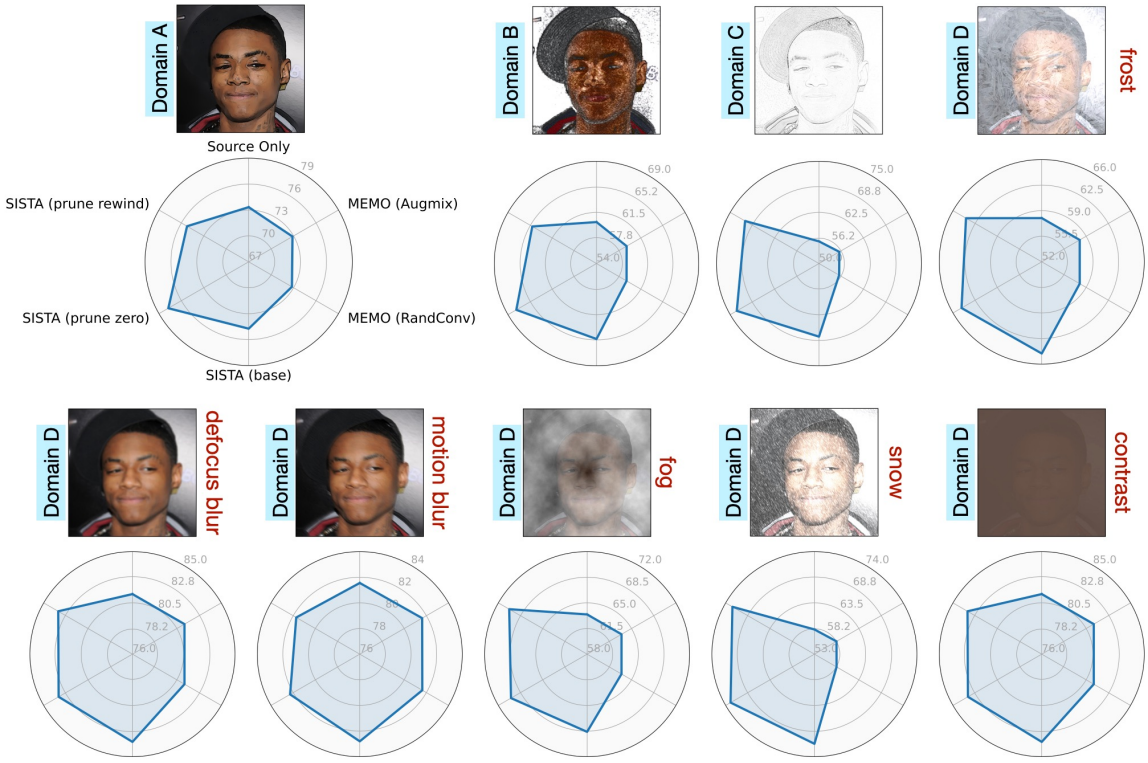


Figure 4. **`SiSTA` significantly improves generalization of face attribute detectors**. We report the 1−shot SFDA performance (Accuracy %) averaged across different face attribute detection tasks for different distribution shifts (Domains A, B & C) and a suite of image corruptions (Domain D). `SiSTA` consistently improves upon the baseline(source-only) and SoTA baseline MEMO in all cases.

10 (Krizhevsky, Hinton, et al. 2009) is also a multiclass classification dataset with 60000 images at 32×32 resolution from 10 different object classes. We use the standard train-test splits for constructing the source and target domain datasets. While we used the StyleGAN-v2 trained on FFHQ faces for our experiments on the CelebA-HQ dataset[1], for AFHQ and CIFAR-10 we obtained the pre-trained StyleGAN2-ADA models[2] from their respective sources; and (iv) DomainNet (Peng et al. 2019a), a

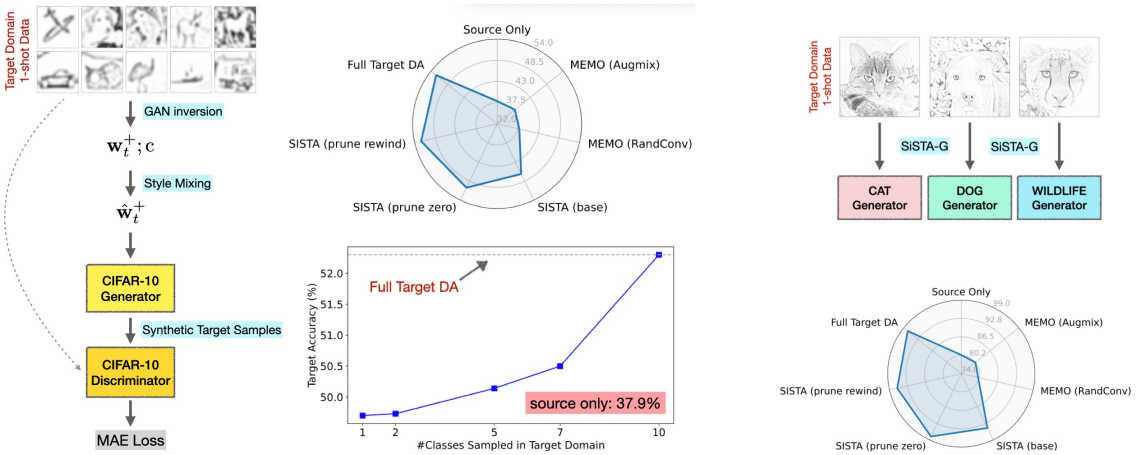
[1] https://github.com/rosinality/stylegan2-pytorch

[2] https://github.com/NVlabs/stylegan2-ada-pytorch

large-scale benchmark comprising 6 domains namely Clipart, Painting, Quickdraw, Sketch, Infograph and Real with each domain consisting of images from 340 categories. For this experiment, we used the state-of-the-art StyleGAN-XL model (Sauer, Schwarz, and Geiger 2022) trained on ImageNet (Russakovsky et al. 2015). Note, we used only the subset of categories from DomainNet that directly overlapped with ImageNet classes. To the best of our knowledge, this is the first work to report adaptation performance with a single target image on DomainNet, and to use ImageNet-scale StyleGAN-XL for data augmentation.

Target Domain Design: To emulate a wide-variety of real-world shifts, we employed standard image manipulation techniques (we will release this new benchmark dataset along with our codes) to construct the following target domains: (i) *Domain A*: We used the *Stylization* technique in OpenCV with $\sigma_s = 40$ and $\sigma_r = 0.2$; (ii) *Domain B*: For this shift, we used the *PencilSketch* technique in OpenCV with $\sigma_s = 40$ and $\sigma_r = 0.04$; (iii) *Domain C*: This challenging domain shift was created by converting each color image to grayscale, and then performing pixel-wise division with a smoothed, inverted grayscale image; and (iv) *Domain D*: This shift was created using a different natural image corruptions from ImageNet-C (Hendrycks and T. Dietterich 2019) typically used for evaluating model robustness. In particular, we used the *imagecorruptions*[3] package for realizing 6 different shifts, namely *contrast, defocus blur, motion blur, fog, frost and snow*. We report our performance across all the domain shifts for the different attribute detection tasks. Given the inherently challenging nature of Domain C, we used that exclusively to evaluate the multi-class classifiers trained on AFHQ and CIFAR-10 datasets. Finally, for DomainNet evaluations we considered *Real photos* as the source domain and used each of the five remaining domains as the target.

[3] https://github.com/bethgelab/imagecorruptions



(a) CIFAR-10. (b) AFHQ.

Figure 5. **Multi-class classification**: (a) Left illustrates `SiSTA-mcG` with class-conditioned GANs, (a) Right shows the performance of `SiSTA`, while the bottom plot studies the performance of `SiSTA` with exposure to only a subset of classes from the target domain. (b) Visualizes our approach for the AFHQ dataset where individual class-specific generators are fine-tuned, and the bottom plot analyses `SiSTA` along with baselines for this challenging dataset.

Evaluation methodology: (a) *Source model training*: The source model F_s is obtained by fine-tuning an ImageNet pre-trained ResNet-50 (He et al. 2016a) with labeled source data. We use a learning rate of $1e-4$, Adam optimizer and train for 30 epochs; (b) *StyleGAN fine-tuning*: We fine-tune G_s for 300 iterations (M in Algorithm 1) using one-target image with learning rate set to $2e-3$ and Adam optimizer with $\beta = 0.99$. These parameters were identified using the CelebA benchmark and we used the same settings for all experiments; (c) *Synthetic data curation*: The size of the synthetic target dataset $\bar{\mathcal{D}}_t$, T, was set to 1000 images in all experiments. Note, in section 2.3.3, we study the impact of this choice. Another important hyperparameter is the choice of GAN layers for style manipulation: (i) layers $8-18$ in StyleGAN-2; (ii) layers $3-8$ in CIFAR-10 GAN; (iii) layers $10-27$ in StyleGAN-XL. This selection was motivated by findings from recent studies on style/content disentanglement in StyleGAN latent spaces (Wu, Lischinski, and Shechtman 2021; Kafri et al. 2021;

Karras, Laine, and Aila 2019). (d) *Choice of pruning ratio*: For all experiments, we set $p = 20\%$ for prune-rewind and $p = 50\%$ for prune-zero strategies. Note, in section 2.3.3, we study the impact of this choice; (e) *SFDA training*: For the NRC algorithm, we set both neighborhood and expanded neighborhood sizes at 5 respectively. Finally, we adapt F_s using SGD with momentum 0.9 and learning rate $1e-3$. All results that we report are computed as an average of 3 independent trials; (f) For evaluation, we report the target accuracy (%) on a held-out test set in each of the target domains.

Baselines: In addition to the vanilla source-only baseline (no adaptation), while there exists a number of test-time adaptation approaches, we perform comparisons to the state-of-the-art online adaptation method, MEMO (Zhang, Levine, and Finn 2021), that enforces prediction consistency between an image and its augmented variants. In particular, we implement MEMO with two popular augmentation strategies namely Augmix and RandConv (Z. Xu et al. 2021). We choose MEMO as the key baseline, since it is already well established that it is superior to other protocols like TENT and TTT. Finally, for comparison, we report the Full Target DA performance as an upper bound, *i.e.*, when the entire target dataset (unlabeled) is used for adaptation.

2.3.2 Findings

Figure 3 illustrates the synthetic data generated for a target domain (*pencil sketch*) using vanilla sampling (or base), *prune-zero* and (*prune-rewind*) strategies. More examples can be found in Figure 8.

`SiSTA` consistently produces superior performance across different distribution shifts.

In Tables 2-10, the performance of `SiSTA` across different domain shifts (A, B, C,

D) on the CelebA-HQ dataset is compared to the baselines for all the 19 attributes. Furthermore, Figure 4 summarizes the average performance (across attributes and multiple trials) for the CelebA-HQ dataset. We see that when compared to the source-only baseline and the state-of-the-art MEMO, `SiSTA` yields average improvements of 4.41%, 7.5%, 17.73% and 5.1% respectively for the four target domains. This improvement can be directly attributed to the efficacy of our proposed augmentations, which enable the SFDA method to learn domain-invariant features when adapting the source classifier.

Additionally, utilizing the proposed activation pruning strategies reveal significant gains under severe shifts over the naïve sampling (base). For example, we see an average improvement of 18% across different attributes in Domain C, when compared to the state-of-the-art MEMO. In particular, we notice that for challenging attributes such as *bangs*, *blond hair*, and *gender*, we obtain striking $26.1\%, 29.6\%, 33.9\%$ improvements over the source-only performance. This illustrates how our pruning strategy can create generic representations that aid in an effective adaptation.

Failure cases: While `SiSTA` is generally very effective, there are a few cases where it does not perform as expected. For example, with the Domain B results in Table 3, we notice that for certain attributes (*5'o clock shadow*, *bald*), we fail to improve over the source-only performance (near-random performance), since it becomes challenging to resolve those attributes under that distribution shift. Additionally, in Domain C, we find that the performance of `SiSTA` (base) is sometimes greater than that of `SiSTA` (prune zero), likely due to the excessive elimination of style information during pruning. While this can be potentially fixed by adjusting the prune ratio or increasing the number of augmented samples (see 2.3.3), this reveals some of the failure scenarios for `SiSTA`.

Table 1. Performance of SiSTA on the five different domains of the DomainNet Dataset. SiSTA consistently improves over the Source Only and MEMO baselines even under such complex domain shifts.

	QuickDraw	Painting	ClipArt	InfoGraph	Sketch
Source only	9.23	62.25	58.55	28.45	43.86
MEMO (Augmix)	8.73	62.20	60.15	28.61	43.86
MEMO (RandConv)	8.04	61.91	59.23	28.02	43.52
SiSTA (base)	11.78	63.53	60.98	31.61	47.54
SiSTA (prune-zero)	**13.12**	63.69	60.98	31.65	**48.12**
SiSTA (prune-rewind)	11.86	**64.05**	**61.02**	**31.8**	46.78
Full Target DA	16.27	68.99	69.55	31.77	55.09

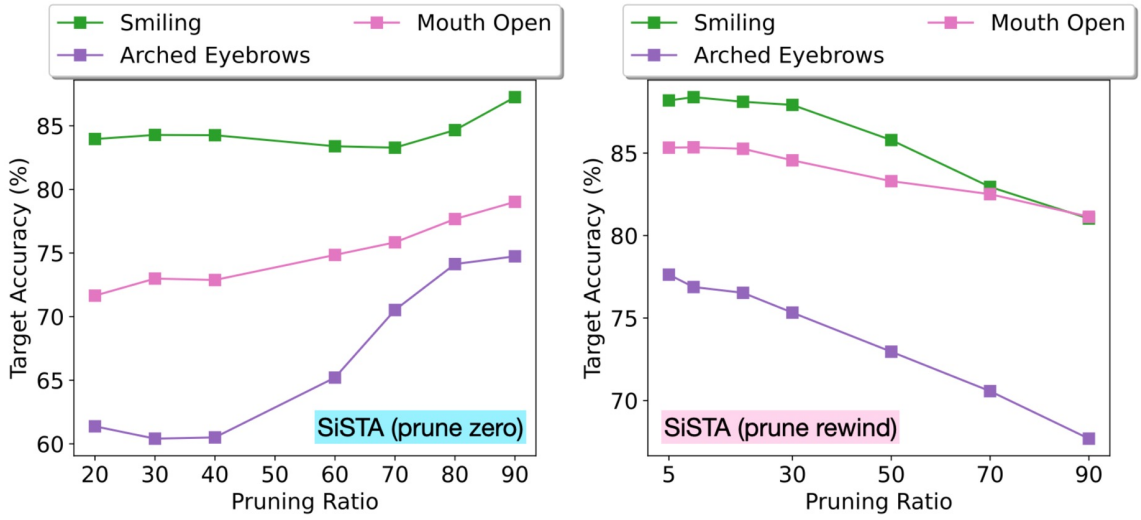
SiSTA can handle natural image corruption. Natural image corruptions mimic domain shifts that are prevalent in real-world settings. Surprisingly, we find that our proposed SiSTA-S protocol is able to fine-tune the GAN even under such image corruptions and lead to apparent gains in the generalization performance. More specifically, we want the emphasize the two challenging corruptions, namely contrast and fog, where the class discriminative features appear to be muted. Even under these corruptions, as showed in Figure 4, SiSTA achieve average performance improvements of 10.14% and 6.52%, respectively.

SiSTA is effective even with class-conditional GANs. In this experiment, we study how SiSTA performs on CIFAR-10 adaptation, when we are provided with a class-conditional StyleGAN. In this case, we use the SiSTA-mcG procedure to perform GAN fine-tuning, which requires the GAN inversion step to identify both the latent code as well as the conditioning variable. As illustrated in Figure 5a, we use $1-$shot examples from each of the 10 classes and synthesize $T = 1000$ augmentations from

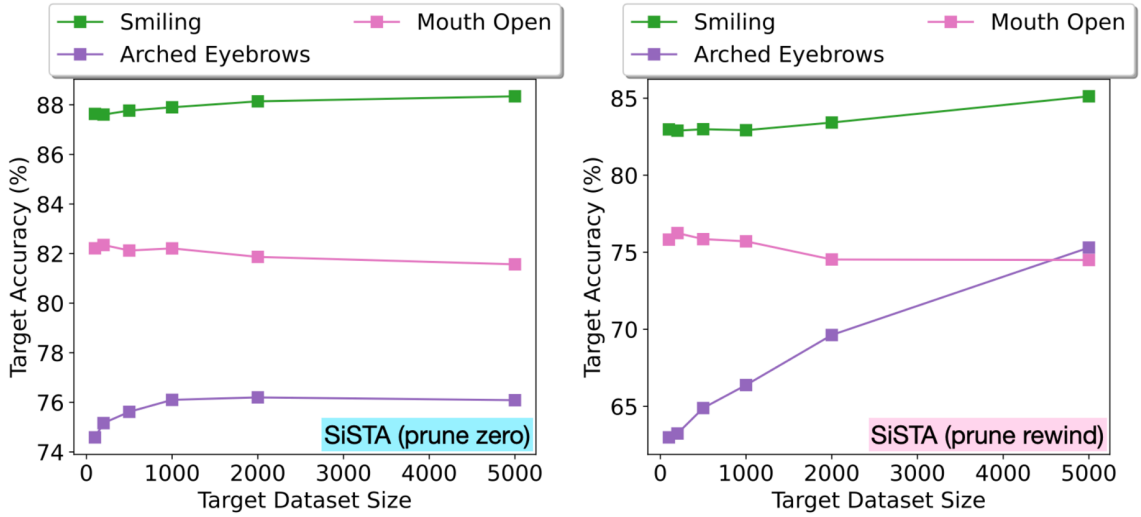
SiSTA. Note, during sampling, we draw from the different classes randomly. We find that, for the challenging Domain C target, SiSTA not only outperforms the baselines by a large margin, but also matches the Full Target DA performance, while using only a single-shot example. Furthermore, as argued in Section 3.3, using single-shot examples from even a subset of classes can be beneficial. To demonstrate this, we varied the number of classes from which target examples are drawn (1 to 10). We find that, even with a single class example, SiSTA provides a large gain of 12.69% over the source-only baseline. As expected, the generalization performance consistently improves as we expose the model to examples from additional classes.

SiSTA can also be used with multiple class-specific GANs. In this study, we examined the performance of SiSTA in a multi-class classification problem with AFHQ, where we assume access to individual generative models for each class. Given the inherent diversity within classes (different breeds of cats or dogs), it is sometimes challenging to train a single StyleGAN for the entire data distribution. In such cases, a separate generative model can be trained on source images from each of the classes. However, the classifier is trained for a $3-$way classification setting. In this case, we perform SiSTA for each GAN independently using its corresponding example. As shown in Figure 5b, we find that, even our base variant achieves 94.53%, outperforming the source-only and baselines by large margins (14%). Our best performance is achieved by *prune-zero* in this setting and it matches Full Target DA.

Even on large scale benchmarks such as DomainNet, SiSTA provides consistent benefits. To study its performance on large-scale benchmarks, we tested SiSTA on DomainNet that comprises a large number of object types and complex distribution shifts (photo, quickdraw, painting, etc.). Given the diversity of objects



(a) Varying prune ratio



(b) Varying T

Figure 6. **Analysis** of varying prune ratio p and the amount of synthetic target domain data T used by `SiSTA`.

in this benchmark, we utilized the state-of-the-art StyleGAN-XL model trained on ImageNet to perform `SiSTA` and studied the single-shot adaptation performance for different target domains (real is the source domain). From Table 1, we find that even on this benchmark, `SiSTA` (prune-zero) convincingly improves upon source only baselines. For example, `SiSTA` provides about 4% improvements for Quickdraw and

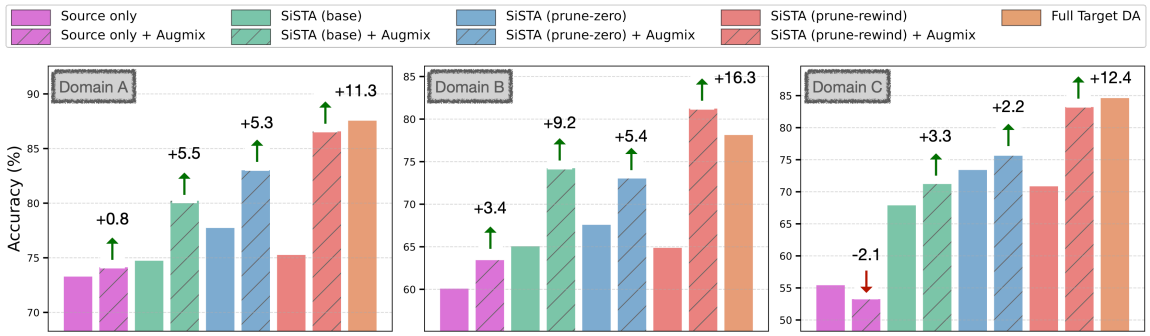


Figure 7. **Effect of Toolbox augmentations on `SiSTA`.** We present the performance of `SiSTA` on Domains A, B, and C of the CelebaHQ dataset when images generated by `SiSTA` are further enhanced with Augmix (Hendrycks et al. 2019). We observe that toolbox augmentations can further improve the performance of `SiSTA`, and in a few cases, `SiSTA` even surpasses the Full Target DA baseline.

Sketch domains. As with the other benchmarks, `SiSTA` is indeed competitive to the Full Target DA baseline.

2.3.3 Analysis of parameter choices

The choice of prune ratio p. We investigate the effect of the choice for p in *prune-zero* and *prune-rewind* using three face attribute detectors (Figure 6a). This parameter influences the degree of generalizability of the synthetic target representations. For *prune-zero*, higher pruning ratios (severe style attenuation), *i.e.*, p between $80 - 90$, are found to significantly enhance performance when compared to lower ones. In the case of *prune-rewind*, on the other hand, p regulates the amount of source mix-up with the target domain. In this scenario, we see that a smaller p performs better, and we recommend to set p between $5 - 20$.

The choice of synthetic data size T. We study the influence of the number of augmentations T by varying it between $100 - 5000$ and studying the performance of *prune-zero* and *prune-rewind* on three attributes, as illustrated in Figure 6b. While

prune-zero performs consistently for different values of T, it only makes limited gains on average as the number of samples increases. On the contrary, we see a significant boost in performance in *prune-rewind* in some of attributes. We remark that *prune-rewind* is a sensitive technique due to the mix-up with the source domain; increasing the number of the synthetic augmentations (along with low p) stabilizes the performance and, in a few cases, even matches the performance of *prune-zero*. Finally, we note that the performance variation across the independent trials is around $< 0.5\%$, thus indicating that the performance is consistent and not sensitive to the sampling process.

Toolbox augmentations can further bolster `SiSTA`. In this study, we investigated the benefits of using sophisticated toolbox augmentations such as Augmix for `SiSTA` as well as for the source only baseline. From Figure 7, we observe a consistent boost in performance for all the three variants of `SiSTA`with average improvements of 6%, 4.2% and almost 13.3% respectively. These results highlight the effective complementary nature of `SiSTA` to toolbox augmentations. Furthermore, it is worth noting that applying Augmix to the source-only methods does not lead to the same level of improvements. This observation is consistent with the findings from (Thopalli, Turaga, and Thiagarajan 2022), which noted that toolbox augmentations alone are insufficient to enhance adaptation performance under real-world distribution shifts.

2.4 Examples of augmentations from `SiSTA`

In Figure 8, we show the augmentations synthesized by `SiSTA` for different domain shifts and StyleGAN models.

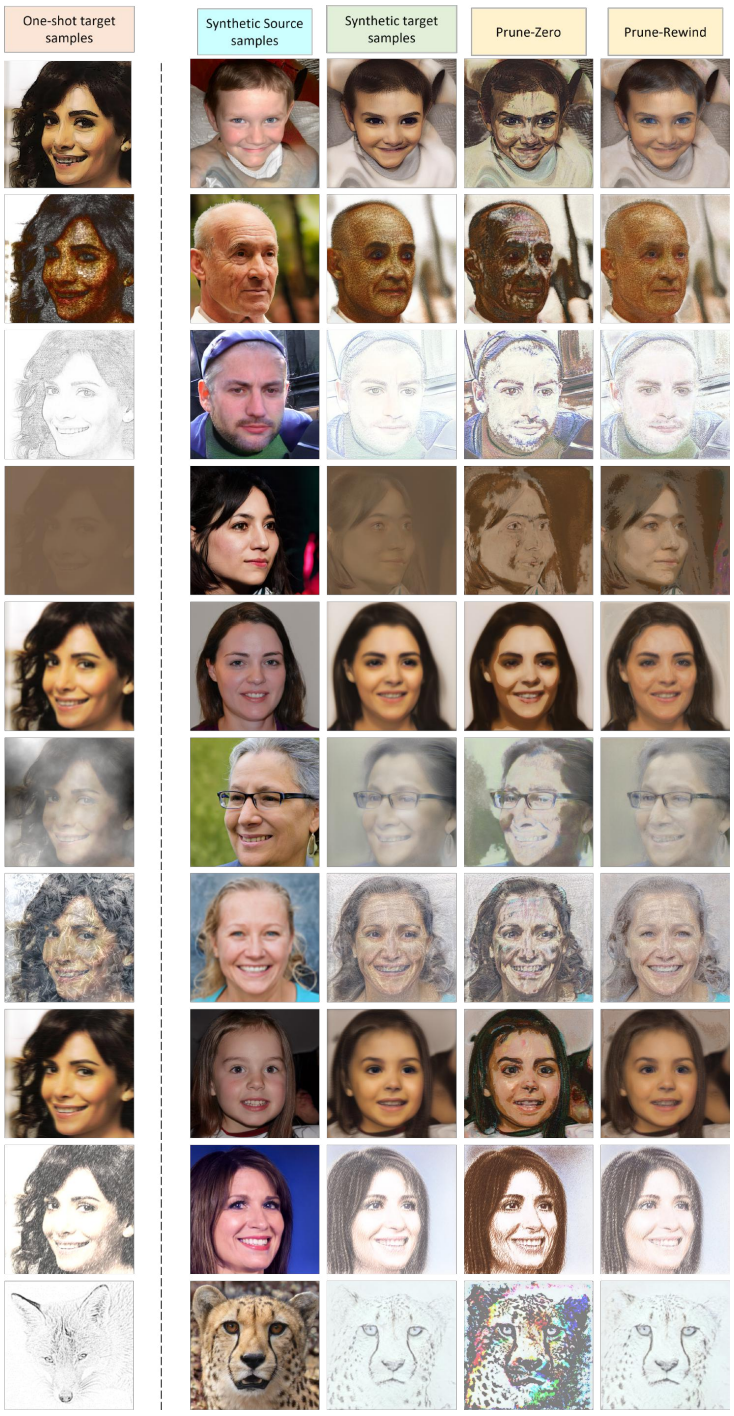


Figure 8. **SiSTA generated augmentations** on random samples drawn from the style space of StyleGAN; The rows 1 to 9 correspond to different domain shifts in CelebA-HQ and row 10 corresponds to AFHQ.

2.5 Detailed results for our CelebA experiments

We provide comprehensive tables for the results discussed in Section 2.3. Tables 2-10 illustrate the performance of source-only, MEMO, and all the three variants of `SiSTA` along with Full target performance.

	5'o clock shadow	Arched eyebrows	Bald	Bangs	Blond hair	Eyeglasses	Makeup	Cheekbones	Gender	Mouth open	Eyes closed	Beard	Sideburns	Smiling	Straight hair	Wavy hair	Earrings	Lipstick	Young
Source only	53.7	69.9	63.4	83.2	55.3	**89.5**	80.7	80.4	93.2	88.2	58.1	**82**	60.2	**89.5**	53.4	68.3	70.9	88.5	64.8
MEMO (Augmix)	53.6	69.9	64.5	81.1	53.8	89.1	79.7	78.6	93.8	87.6	57.9	80.8	59.6	89.4	52.5	70.5	68.6	88.1	65.1
MEMO (Randconv)	53.7	69.6	64.5	81	53.7	89.1	79.5	78.4	93.9	87.6	57.9	80.8	59.5	89.3	52.5	70.2	68.6	88	65
SiSTA (base)	52.8	74.6	**77**	80	85.2	69.8	87.2	72.8	95.1	91.2	55.2	69.8	58.3	84.4	57	79.1	71.3	**90.1**	**69.1**
SiSTA (prune-zero)	**55.2**	**78.2**	76.3	**87.1**	**87.6**	81.5	**88.1**	**81.2**	**95.5**	**91.7**	**60.4**	70.8	**61.1**	89.2	**59.3**	**79.5**	**76.2**	89.6	68.6
SiSTA (prune-rewind)	53.1	76.6	70.1	85.6	83	78.2	87.1	76	95.2	91.6	57.8	67.5	58.5	87.3	59.2	78.6	74.2	89.3	60.6
Full target DA	87	81.9	92.3	93.5	90.1	97.3	89.3	87.1	97.4	92.7	72.5	91.5	93	92.6	74.5	80.6	82.5	92.3	75.2

Table 2. Performance of `SiSTA` on Domain A of the CelebA dataset.

	5'o clock shadow	Arched eyebrows	Bald	Bangs	Blond hair	Eyeglasses	Makeup	Cheekbones	Gender	Mouth open	Eyes closed	Beard	Sideburns	Smiling	Straight hair	Wavy hair	Earrings	Lipstick	Young
Source only	50	51	50.5	67.2	50	74.2	54.2	54.6	80.2	78.6	52.1	**63.9**	**54**	76.9	50.1	65	50.4	63.3	55.5
MEMO (Augmix)	50	51.2	50.5	64.5	50	74.1	52.1	52.4	81.1	79	51.2	63	50.8	73.2	50	65.5	50.2	58.6	55.6
MEMO (Randconv)	50	51.2	50.5	64.5	50	73.9	52.1	52.3	81.2	79	51.2	62.9	50.8	73.1	50	65.6	50.2	58.5	55.7
SiSTA (base)	50	73	50.2	83.3	50.5	67.8	77.6	56.3	86.5	82.5	56.7	56.1	50.1	77	51.7	72.6	**56.3**	**80**	58.1
SiSTA (prune-zero)	50.1	**73.9**	51.1	**86.7**	51.4	**75.8**	**79.9**	**67.2**	**88.7**	**84.4**	**58.3**	58.1	50.2	**85.4**	**53.8**	**74**	54.8	79.8	**60.5**
SiSTA (prune-rewind)	50	73.4	50	84.7	50.2	75.2	75.5	57.1	85.9	82.9	54	54.5	50.1	78	52.7	72.8	**56.3**	73	56.3
Full target DA	71.6	71.7	72.6	89.9	58.4	94.2	81.9	78.5	92.2	88	63.9	84.3	83	88.4	68.6	71	68.6	86.7	71.2

Table 3. Performance of `SiSTA` on Domain B of the CelebA dataset.

2.6 Conclusion

In this chapter, we explored the use of generative augmentations for test-time adaptation, when only a single-shot target is available. Through a combination of

	5'o clock shadow	Arched eyebrows	Bald	Bangs	Blond hair	Eyeglasses	Makeup	Cheekbones	Gender	Mouth open	Eyes closed	Beard	Sideburns	Smiling	Straight hair	Wavy hair	Earrings	Lipstick	Young
Source only	50	52.8	50.1	58.2	50.5	63.8	56.5	50.2	58.3	58.9	50	51.3	50.5	64	52	59.9	51.8	71.6	52.7
MEMO (Augmix)	50	53.6	50.2	61.6	50.5	66.6	55.5	50.1	56.1	60.4	50	50.8	50.4	65.8	52	59.2	51.7	72.3	52
MEMO (Randconv)	50	53.6	50.2	61.6	50.5	66.6	55.4	50.1	56	60.4	50	50.7	50.4	65.5	52	59.1	51.7	72.4	52
SiSTA (base)	53.2	65.3	**64.7**	80	77.9	69.4	54.5	71.2	91.8	71.4	**59.1**	66.6	53.2	79.2	54.7	77.3	57.8	78.8	63.7
SiSTA (prune-zero)	**58**	**74.7**	64.1	82.6	77.1	**82.7**	**80.7**	**77.2**	88.3	**78.2**	56.3	**68.2**	**55.3**	**86.7**	**68.5**	74.3	**62.8**	**86.5**	67.6
SiSTA (prune-rewind)	53.1	69.7	63.5	**84.3**	**80.1**	79.9	62.1	69.7	**92.2**	**78.2**	54.4	65	53.7	84.4	57.3	**78.5**	58.2	**86.5**	**74.5**
Full target DA	83.1	80.5	92	93	84.2	96.7	83.8	80.8	95.7	87.6	66.9	90	93.2	89.2	69.9	77.5	76.6	89.5	77.5

Table 4. Performance of `SiSTA` on Domain C of the CelebA dataset.

	5'o clock shadow	Arched eyebrows	Bald	Bangs	Blond hair	Eyeglasses	Makeup	Cheekbones	Gender	Mouth open	Eyes closed	Beard	Sideburns	Smiling	Straight hair	Wavy hair	Earrings	Lipstick	Young
Source only	64.5	79	82.5	90	87.4	91	90.4	87.8	97.2	92	64.5	79.7	63.4	93	68.8	79.9	65.7	92.6	74.9
MEMO (Augmix)	63.2	78.1	87.5	88	87.1	**91.3**	90.6	**89.8**	**97.8**	90.8	65.3	77.4	62	**92.9**	70.6	80.8	63.9	91	75.5
MEMO (Randconv)	63.2	78.1	87.5	87.5	87.1	**91.3**	90.6	**89.8**	**97.8**	90.8	65.3	77.4	62	**92.9**	70.6	81	63.7	91	75.3
SiSTA (base)	**85.6**	80	**88.9**	88.9	91.2	76.9	89.8	79	95.3	91.5	**65.6**	**91.4**	**89.3**	87.5	65.2	**82.4**	68.2	91.9	**81.9**
SiSTA (prune-zero)	85.1	79.5	85.1	90.3	**92.8**	83.3	**90.7**	82.4	96.4	90.7	63.8	89.7	76.7	89.9	**73.7**	81.5	69.3	**92.2**	73.3
SiSTA (prune-rewind)	78.2	**81.5**	85.3	**92.3**	92.5	83.4	90.5	81.7	97.2	**92.7**	64.2	87.7	77.3	90.7	71.1	82.1	**71.1**	**92.2**	75.5
Full target DA	89.4	83	96.1	94	92.9	97.1	90.7	88	97.8	93.7	74.4	93.3	94.1	93.3	76.9	82.4	84.5	92.6	83.1

Table 5. Performance of `SiSTA` on Domain D (Defocus blur) of the CelebA dataset.

	5'o clock shadow	Arched eyebrows	Bald	Bangs	Blond hair	Eyeglasses	Makeup	Cheekbones	Gender	Mouth open	Eyes closed	Beard	Sideburns	Smiling	Straight hair	Wavy hair	Earrings	Lipstick	Young
Source only	71.4	**79.7**	79.5	88.3	88.9	91.5	89.7	87.1	97.6	91.6	69.6	80.5	65.7	92.9	72.5	73.9	62.2	92.2	74.8
MEMO (Augmix)	73	78.6	73.7	88.3	88.8	**91.8**	91.9	**88.5**	**97.5**	92	**70.7**	80.8	63	**93.1**	**73.5**	75	62.2	92.6	75.5
MEMO (Randconv)	73	78.6	73.7	88.3	88.8	**91.8**	**92**	**88.5**	**97.5**	92.1	**70.7**	80.8	63	**93.1**	**73.5**	75	62.2	**92.7**	75.5
SiSTA (base)	**79.8**	74.7	**89.8**	89.3	**93.6**	78.2	89.6	79.5	94.4	92.2	67.4	**87.8**	**73.1**	87.9	69.7	**81.5**	**71**	92	**82**
SiSTA (prune-zero)	74	75.4	87.1	92.1	**93.6**	86.9	90.6	83.7	96.5	91.4	66.3	78.6	63	90.8	72.9	81.2	70.9	92.4	76.3
SiSTA (prune-rewind)	70.7	76.1	85.9	**92.5**	**93.6**	85.5	90	81.2	96.2	**92.8**	65.9	79.7	64.9	89.9	72.5	80.4	68.9	92.2	73.9
Full target DA	90.1	82.8	96.7	93.8	93.2	98.1	90.8	88.2	97.9	93.7	72	94.9	94.6	93.2	75.7	82.6	85.4	92.9	84.2

Table 6. Performance of `SiSTA` on Domain D (Motion blur) of the CelebA dataset.

	5'o clock shadow	Arched eyebrows	Bald	Bangs	Blond hair	Eyeglasses	Makeup	Cheekbones	Gender	Mouth open	Eyes closed	Beard	Sideburns	Smiling	Straight hair	Wavy hair	Earrings	Lipstick	Young
Source only	59.5	52.5	**71.9**	59.9	51.9	88.1	50	57.6	78.3	79.6	50.6	82	63	77.2	**53.8**	63.6	52.5	50.6	62.5
MEMO (Augmix)	**62.6**	52.5	60.9	61.4	52.4	83	50	57.9	78.2	78.9	50.6	81.3	**64.1**	77.1	52	62.3	53.1	50.5	61.1
MEMO (Randconv)	**62.6**	52.4	60.9	61.1	52.4	83	50	57.9	78.3	78.9	50.6	81.3	63.5	77	51.6	62.3	53.1	50.5	61.1
SiSTA (base)	57.3	56.3	**71.9**	77.9	57.4	80.6	60.7	68.2	75.2	84.2	**57.1**	**84.9**	63	76.8	51.8	74.8	62.3	73.5	69.6
SiSTA (prune-zero)	54.1	57.3	70.8	80.6	**58.9**	**89**	63.6	**77**	**81.1**	82.8	56.4	73.9	55.4	**85.3**	53.7	**76.2**	**63.8**	78.2	**71.7**
SiSTA (prune-rewind)	54.3	**58.5**	68.8	**84.3**	53.6	87.3	**69.4**	75.9	78.4	**85.8**	56.1	80.9	60	81.5	52.1	74.8	62.2	**80.8**	70.6
Full target DA	86.9	78.8	80	90.6	90	97.8	85.9	82.9	94.6	92.9	72.6	92.6	92.5	89	70.6	78.3	84.2	89.6	76

Table 7. Performance of `SiSTA` on Domain D (Fog) of the CelebA dataset.

	5'o clock shadow	Arched eyebrows	Bald	Bangs	Blond hair	Eyeglasses	Makeup	Cheekbones	Gender	Mouth open	Eyes closed	Beard	Sideburns	Smiling	Straight hair	Wavy hair	Earrings	Lipstick	Young
Source only	51.1	51.5	54.6	55.8	51.3	70.5	50.1	53.5	75.5	72.9	50.8	68.6	56.3	66.7	50.2	64.6	51.7	51.5	54.7
MEMO (Augmix)	50.3	51.4	55.6	55.8	51.4	**72**	50.2	53.9	75.9	74.3	51.1	68.3	56.4	67.2	50	64.3	51.2	51.1	53.9
MEMO (Randconv)	50.3	51.4	55.6	55.8	51.4	71	50.2	53.9	75.9	74.4	51.1	68.4	56.4	67.2	50	64	51.2	51.1	54
SiSTA (base)	51	65.2	58.7	60.4	59.8	62.2	76	58.5	**82.6**	79.5	**57.8**	**69.2**	51.9	74.4	54.8	**66.4**	**62.1**	**77.9**	**58.2**
SiSTA (prune-zero)	50.5	**66.3**	**59.3**	59.4	**70.9**	65.3	75.6	**66.8**	80.4	76.4	57.7	64.1	50.3	**78.7**	**56.3**	58.7	61.1	73.6	57.1
SiSTA (prune-rewind)	50.3	65.6	55.6	**61.2**	61	65.4	**76.5**	60.4	82.3	**80.2**	56.2	64.6	50.4	76	54.9	61.7	61.8	77	53.9
Full target DA	67.6	68.8	65.5	75.6	75.9	90.7	78.6	76.7	88	89.3	61.9	74.2	61.2	86.2	61.3	60.2	67.2	84.9	62.2

Table 8. Performance of `SiSTA` on Domain D (Frost) of the CelebA dataset.

	5'o clock shadow	Arched eyebrows	Bald	Bangs	Blond hair	Eyeglasses	Makeup	Cheekbones	Gender	Mouth open	Eyes closed	Beard	Sideburns	Smiling	Straight hair	Wavy hair	Earrings	Lipstick	Young
Source only	50.1	52.5	50.9	59.6	50.2	73.9	51.9	50.1	76.9	66.4	50.1	70.6	57.7	62.9	50.8	61.6	51.2	54	61.8
MEMO (Augmix)	50	52.7	50	60	50	73.5	51.5	50	77	69.6	50	70.2	58.7	64.4	50.7	61	51.3	54.3	61.7
MEMO (Randconv)	50	52.7	50	60	50	73.5	51.3	50	77	69.7	50	70	58.8	64.7	50.8	60.9	51.3	54.3	61.6
SiSTA (base)	62	64.2	60.9	67.2	79.1	82.6	79.4	65	83.4	77.4	**68.7**	**82.1**	**60.3**	75.4	54.7	75.6	**67.1**	84.3	66.6
SiSTA (prune-zero)	**62.4**	63.2	**61.4**	70.7	**87.2**	**87.8**	79.5	**68.8**	86.2	75.7	67.8	81.4	59.7	79.4	**57.9**	**76.1**	64.9	86.4	**67.4**
SiSTA (prune-rewind)	59.4	**67.5**	57.5	**73.6**	79.4	86.2	**83.3**	65.9	**86.7**	**79.3**	66.7	81.2	58.7	**79.7**	55.8	74.9	67	**87.1**	65.9
Full target DA	76.32	77.68	66.79	82.68	85.69	94.96	84.32	77.87	89.45	83	70.09	85.44	83.71	85.55	62.11	72.93	79.14	84.89	65.89

Table 9. Performance of `SiSTA` on Domain D (Snow) of the CelebA dataset.

	5'o clock shadow	Arched eyebrows	Bald	Bangs	Blond hair	Eyeglasses	Makeup	Cheekbones	Gender	Mouth open	Eyes closed	Beard	Sideburns	Smiling	Straight hair	Wavy hair	Earrings	Lipstick	Young
Source only	50	50	50.4	53	53.4	51.5	50	51.2	69.7	54.5	50	58.9	50	59.4	50.5	50.8	50	56.5	61.6
MEMO (Augmix)	50	50	52.6	51.8	51.9	52.1	50	51.1	68.9	54.5	50	**58.8**	50	58.3	49.9	50.6	50	55.9	57.3
MEMO (Randconv)	50	50	52.6	51.8	51.4	52.1	50	51.1	69	54.6	50	**58.8**	50	58.3	49.9	50.6	50	55.7	58.1
SiSTA (base)	50	60.1	54.1	70.3	66.7	50.3	72	65.5	**83.5**	75.3	50.8	52.5	50.6	74.4	51.7	**70.6**	51.2	**77.3**	**62.5**
SiSTA (prune-zero)	50	**65.7**	**58.7**	**76.4**	**75.6**	51.1	**80.1**	**73.7**	74.2	73.2	50.3	51.1	50.2	**82.9**	54.9	67.1	52.2	72	57.8
SiSTA (prune-rewind)	50	63.4	55	72.8	72	51	76	67	81.6	**76.9**	50.5	52.4	50.2	78.5	**55**	69.8	50.5	76.4	61.5
Full target DA	63.2	76.7	65.56	69.31	76.7	92.1	76	76.3	86.4	89.2	58.8	73.1	71.8	87.1	57.1	68.5	73.7	80.5	62.3

Table 10. Performance of `SiSTA` on Domain D (Contrast) of the CelebA dataset.

StyleGAN fine-tuning and novel sampling strategies, we were able to curate synthetic target datasets that effectively reflect the characteristics of any target domain. We showed that the proposed approach is effective in multi-class classification using both class-conditioned as well as multiple class-specific GANs. Our future work includes theoretically understanding the behavior of different pruning techniques and extending our approach beyond classifier adaptation.

Chapter 3

STYLEGAN-V2 BASED INVERSION FOR OUT-OF-DISTRIBUTION IMAGES

In the past few years, generative adversarial networks (GANs) (Goodfellow et al. 2014) have been shown to produce high-quality, photo-realistic images in a variety of image synthesis and manipulation tasks (Karras, Laine, and Aila 2019; Härkönen et al. 2020b; Brock, Donahue, and Simonyan 2019; G. Song et al. 2021). In particular, the StyleGAN-v2 architecture and its variants (Karras, Laine, and Aila 2019; Karras et al. 2020; Karras et al. 2021) have been used to synthesize very high resolution images. At a basic level, StyleGAN-v2 learns to transform a latent vector $\mathbf{z} \in \mathcal{Z} \subseteq \mathbb{R}^{512}$ to an intermediate latent code $\mathbf{w} \in \mathcal{W} \subseteq \mathbb{R}^{512}$ through a mapping function (projection head) f, which is then used to synthesize images.

The continued progress in training GANs has led to a surge in techniques that can leverage deep generators as priors for ill-posed image inversion problems (Bora et al. 2017; Anirudh et al. 2019; Daras et al. 2021). In this context, the problem of accurately embedding a given image onto the latent space, often referred to as GAN inversion, has gained significant research interest (Abdal, Qin, and Wonka 2019, 2020; Daras et al. 2021; P. Zhu et al. 2020; Wulff and Torralba 2020; Kang, Kim, and Cho 2021). Furthermore, it has been demonstrated that the rich semantic information encoded in the latent space of a pre-trained StyleGAN allows seamless editing of images through controlled latent code manipulations (P. Zhu et al. 2020; Karras, Laine, and Aila 2019).

Broadly, existing approaches for StyleGAN-based inversion perform a careful selection of the latent space for optimization ($\mathcal{Z}, \mathcal{W}$ and their variants) and regularization

techniques. Though existing inversion strategies have been successful with images that are similar to the data used for training the StyleGAN (e.g., FFHQ faces), embedding out-of-distribution images (e.g., an X-ray) onto the latent space is known to be very challenging. To this end, Abdal *et al.* (Abdal, Qin, and Wonka 2019) demonstrated that it is possible to invert out-of-domain images, e.g., car images, onto the $\mathcal{W}+$ space of a pre-trained StyleGAN. However, the perceptual quality of the reconstructed images became poorer when non-face images were considered and this can be attributed to the mismatch between the latent space prior and the OOD data.

Proposed Work. In this chapter, we introduce SPHInX (StyleGAN with Projection Heads for Inverting X), an inversion approach for accurately embedding any arbitrary OOD image onto the latent space of StyleGAN-v2 (Karras et al. 2020). We systematically study the behavior of different existing latent space optimization strategies, using a broad suite of non-face image datasets, and show that they are not very effective at reconstructing OOD images. We make a critical finding that, by redesigning the projection head that maps between $\mathcal{Z}+$ and $\mathcal{W}+$, such that the style latent variables corresponding to different intermediate layers in the generator architecture are decoupled, one can significantly improve the inverse optimization process. In a nutshell, SPHInX improves OOD image embedding by: (a) replacing the existing mapping function f with a style projection head $\mathcal{P}_s$;(b) introducing a content projection head $\mathcal{P}_c$ and leveraging the noise latent variables $\mathcal{B}$; and (c) adopting a novel training strategy that enforces $\mathcal{P}_s$ to consistently produce a meaningful solution in the $\mathcal{W}+$ latent space for any realization from $P(\mathcal{Z}+)$, which in turn induces a robust estimate of $P(\mathcal{W}+)$.

Contributions. (a) A new approach, SPHInX, for inverting OOD images onto the StyleGAN latent space; (b) Design of a style projection head that maps between $\mathcal{Z}+$

and $\mathcal{W}+$, to improve inversion with OOD data; (c) Novel training strategy that induces a robust local neighborhood in $\mathcal{W}+$ for a given image; (d) Extensive empirical studies on non-face image data to demonstrate the efficacy of SPHInX in reconstruction and solving inverse problems (denoising, super-resolution and compressed sensing); (e) Application of SPHInX to perform simultaneous inversion and attribute discovery; (f) Our codes are can be accessed at https://anonymous.4open.science/r/SPHInX.

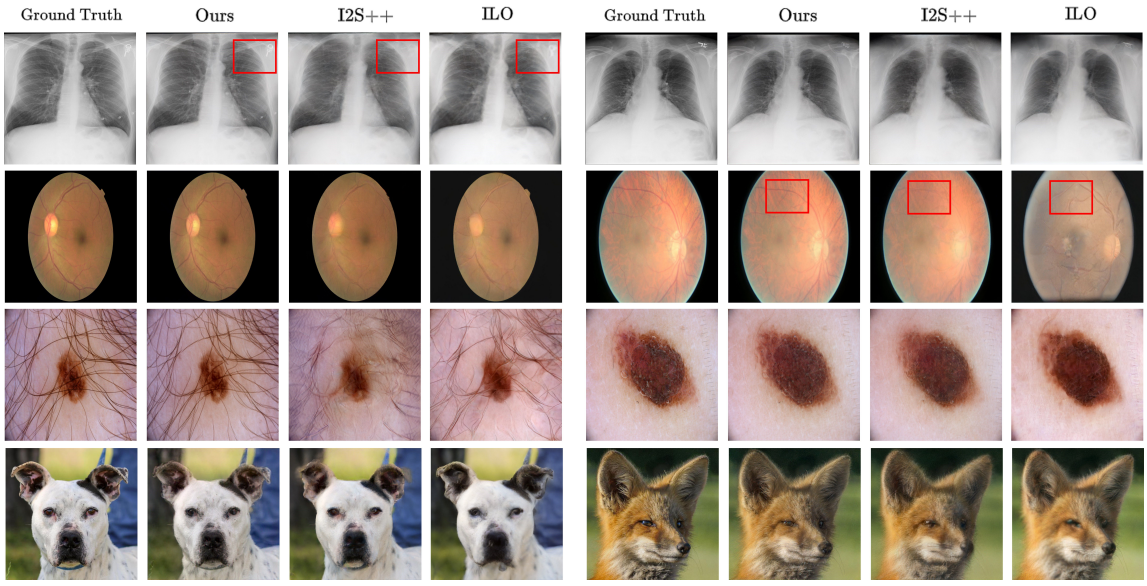


Figure 9. **Comparing out-of-distribution reconstruction of SPHInX with I2S++ (Abdal, Qin, and Wonka 2020) and ILO (Daras et al. 2021)**. Our approach accurately inverts images onto the StyleGAN-v2 latent space across a variety of datasets.

3.1 Background

GAN Inversion. This refers to the ill-posed problem of inferring a latent code or an embedding $\mathbf{z}$ for an image in the latent space of a pre-trained generative model $\mathcal{G}$. Such an inversion technique can be utilized for semantic manipulation or solving restoration

Table 11. StyleGAN-based inversion involves optimizing the latent spaces $\mathcal{Z}+, \mathcal{S}$, and $\mathcal{B}$ in different combinations. Various optimization and regularization strategies have been proposed in the literature to improve the efficacy of this inversion process.

Method	**Opt. Space**	**Add. Regularization Strategy**
PGD (Karras, Laine, and Aila 2019)	$\mathcal{Z}$	-
PULSE (Menon et al. 2020)	$\mathcal{Z}+$	Latent space search under high-dimensional Gaussian prior
ILO (Daras et al. 2021)	$(\mathcal{Z}+, \mathcal{S}, \mathcal{B})$	ℓ_1-ball constraint on the manifold induced by the previous layer
I2S (Abdal, Qin, and Wonka 2019)	$\mathcal{W}+$	-
Zhu et al. (P. Zhu et al. 2020)	$\mathcal{W}+$	PCA whitening on transformed $\mathcal{W}+$ space (referred as $\mathcal{P}$)
IDInvert (J. Zhu et al. 2020)	$\mathcal{W}+$	In-domain regularization using domain-guided encoder
PIE (Tewari, Elgharib, Bernard, et al. 2020)	$\mathcal{W}+$	Hierarchical non-linear optimization
Wulff et al. (Wulff and Torralba 2020)	$\mathcal{W}+$	Statistical priors on $\mathcal{W}+$ space
StyleFlow (Abdal et al. 2021)	$\mathcal{W}+$	-
StyleRig (Tewari, Elgharib, Bharaj, et al. 2020)	$\mathcal{W}+$	Self-supervised two-way cycle consistency loss
I2S++ (Abdal, Qin, and Wonka 2020)	$(\mathcal{W}+, \mathcal{B})$	-
BDInvert (Kang, Kim, and Cho 2021)	$(\mathcal{W}+, \mathcal{S})$	$\mathcal{P}$ whitening and semantic consistency regularization

tasks such as in-painting and compressed sensing (Bora et al. 2017). Projected gradient descent (PGD) (Abdal, Qin, and Wonka 2019; Anirudh et al. 2019; Abdal, Qin, and Wonka 2019; Shah and Hegde 2018; Yeh et al. 2017; Raj, Li, and Bresler 2019; Mitra

et al. 2023) is a commonly adopted strategy, which optimizes for a latent vector that minimizes a discrepancy $\mathcal{L}(.,.)$ between the generated image $\mathcal{G}(\mathbf{z})$ and the given observation I. Mathematically,

$$\mathbf{z}^* = \arg\min_{\mathbf{z}} \ \mathcal{L}(G(\mathbf{z}), I) + \mathcal{R}(.), \tag{3.1}$$

where $\mathcal{R}(.)$ is an additional regularizer. Common choices for $\mathcal{L}$ are: (i) pixel-wise mean squared error ($\mathcal{L}_{\text{MSE}}$) and (or) (ii) learned perceptual image patch similarity ($\mathcal{L}_{\text{LPIPS}}$) (R. Zhang et al. 2018) which is a perceptual similarity metric based on deep network activations (VGG-16 (Simonyan and Zisserman 2015)):

$$\mathcal{L}_{\text{LPIPS}} = \sum_{\ell} \frac{1}{H_\ell W_\ell} \sum_{h,w,c} w_{\ell c} (\Psi^\ell_{I_{hwc}} - \Psi^\ell_{I'_{hwc}}))^2, \tag{3.2}$$

where (H_ℓ, W_ℓ) denotes the spatial size in layer ℓ and Ψ^ℓ denotes the ℓ^{th} latent layer of the adopted classifier. Further, $w_{\ell c}$ corresponds to the channel-level scaling vector, and I, I' are the images being compared.

StyleGAN Preliminaries. At its core, StyleGAN (Karras, Laine, and Aila 2019; Karras et al. 2017a) relies on a mapping network f that transforms an input latent code $\mathbf{z} \in \mathbb{R}^{512}$ sampled from a Gaussian prior $P(\mathcal{Z})$ to a disentangled intermediate latent code $\mathbf{w} \in \mathbb{R}^{512} \in \mathcal{W}$. The latent code $\mathbf{w}$ is then repeated $N_\ell = 18$ times and passed to the AdaIn blockKarras, Laine, and Aila 2019; Huang and Belongie 2017 of each of the layers in $\mathcal{G}$. Differing from conventional generative models (Goodfellow et al. 2014; Radford, Metz, and Chintala 2016), instead of directly passing $\mathbf{z}$ to the first layer, StyleGAN uses a constant input $\mathbf{s} \in \mathbb{R}^{4\times4\times512}$ (initially drawn at random from a Gaussian prior $P(\mathcal{S})$) which is progressively transformed in every layer with increasing resolution to synthesize the images. Additionally, StyleGAN employs a set of noise inputs sampled independently from a Gaussian prior $P(\mathcal{B})$, in every layer to improve the overall textural quality.

StyleGAN-based Inversion. Since pre-trained StyleGAN can be effectively leveraged as a prior for ill-posed image recovery and semantic editing, several StyleGAN-specific inversion studies have emerged recently (Abdal, Qin, and Wonka 2019, 2020; Daras et al. 2021; Menon et al. 2020; P. Zhu et al. 2020; Wulff and Torralba 2020). While performing StyleGAN-based inversion, the choice of the latent space ($\mathcal{Z}$, $\mathcal{W}$ and their variants) along with additional regularization techniques adopted become critical. Chapter 11 provides a comprehensive list of StyleGAN-based inversion strategies, along with their choice of latent space optimization and regularization.

Abdal *et al.* (Abdal, Qin, and Wonka 2019) (*Image2StyleGAN*, or shortly I2S), first investigated the efficacy of inverting an image onto the intermediate $\mathcal{W}$ space of StyleGAN. They made a crucial observation that the reconstruction quality can be significantly improved by optimizing with an extended intermediate latent space $\mathcal{W}+ \subseteq \mathbb{R}^{N_\ell \times 512}$, where every $\mathbf{w}^+ \in \mathcal{W}+$ was obtained by stacking N_ℓ realizations from $P(\mathcal{Z})$ using the mapping f. In addition, they demonstrated that $\mathcal{W}+$ offers a higher degree of freedom to guide the inversion compared to the $\mathcal{W}$. As an extension, Abdal *et al.* (Abdal, Qin, and Wonka 2020) identified that images can be reconstructed with improved granularity by optimizing the noise space $\mathcal{B}$ along with $\mathcal{W}+$ (*Image2StyleGAN++*, or shortly I2S++). Wulff *et al.* (Wulff and Torralba 2020) and Zhu *et al.* (P. Zhu et al. 2020), on the other hand, introduced statistical priors over $\mathcal{W}+$ to better control and regularize the inversion. Recently, Daras *et al.* (Daras et al. 2021) proposed an optimization strategy that progressively included latent variables from different layers of StyleGAN and optimized for latent codes in $\mathcal{Z}+$ that lie within an ℓ_1-ball around the manifold induced by the previous layer (*Intermediate Layer Optimization*, or shortly ILO). Here, every $\mathbf{z}^+ \in \mathcal{Z}+$ was obtained by stacking N_ℓ realizations from $P(\mathcal{Z})$. However, these studies extensively focus on in-domain

images (Karras et al. 2020). Although Kang *et al.* (BDInvert (Kang, Kim, and Cho 2021)) proposed out-of-range face image inversion by introducing an encoder-based regularization on the generator feature maps, their approach is not applicable for OOD images, without access to domain-specific encoders. We address this crucial gap and investigate StyleGAN-based OOD image inversion.

3.2 Proposed Approach

3.2.1 Motivation

While existing approaches can effectively invert images by optimizing in the extended $\mathcal{W}+$ latent space (along with semantic and noise latent variables $\mathcal{S}$ and $\mathcal{B}$), their performance with OOD image data, *e.g.*, non-faces, is found to be sub-optimal (Abdal, Qin, and Wonka 2019). Given that the $\mathcal{W}+$ latent space contains rich semantic information about in-domain images (for e.g., faces), the latent codes need to be significantly altered in order to accurately reconstruct OOD images devoid of in-domain artifacts. Despite the large number of degrees of freedom, the effectiveness of $\mathcal{W}+$ optimization could be limited by the initialization provided by the mapping network f, which serves as a prior for in-domain specific styles (Karras, Laine, and Aila 2019). As illustrated using a perceptual quality metric (LPIPS defined in Eq. 3.2) in Figure 10, the solution obtained by optimizing in the collection of latent spaces $(\mathcal{W}+, \mathcal{S}, \mathcal{B})$ of StyleGAN-v2 is highly non-robust for a CXR image. Even minor perturbations to the solution (additive noise to achieve a target SNR in each of the latent spaces) introduces irrelevant face-like features into the X-ray image. This naturally motivates the need for novel optimization strategies that can provide

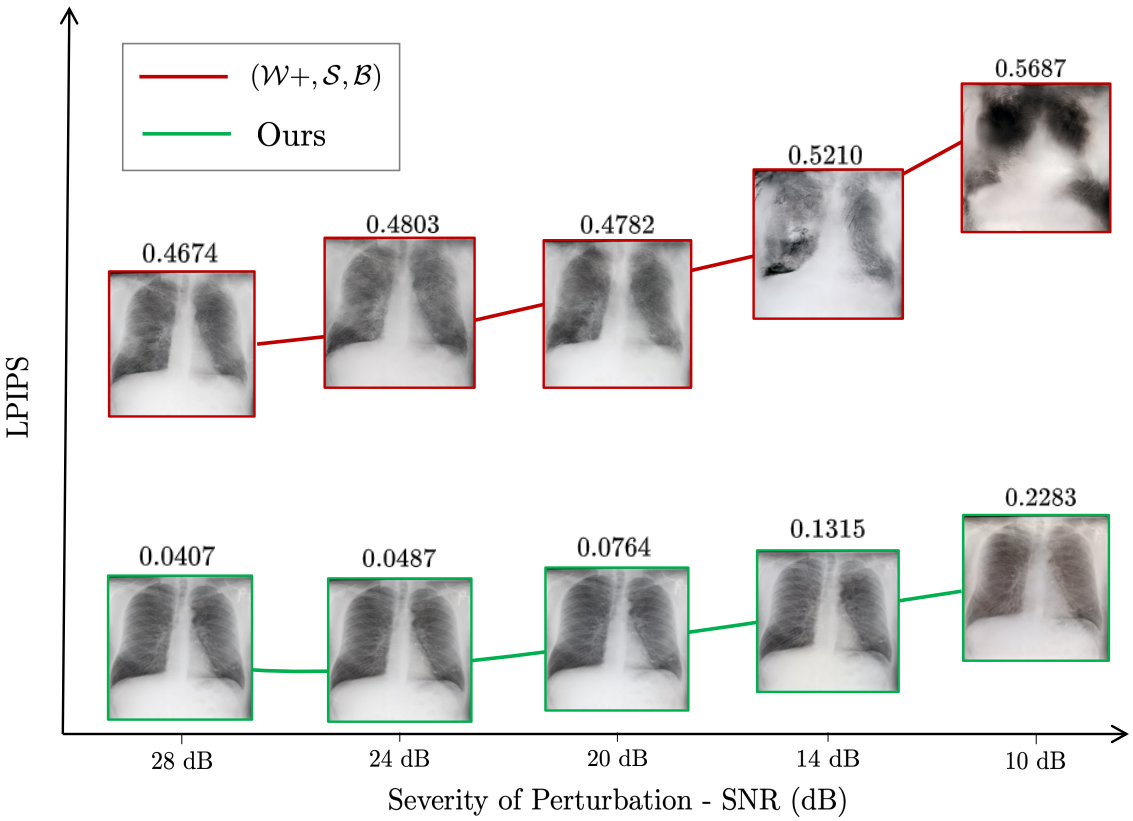


Figure 10. **Robustness of GAN inversion methods under latent space perturbations**. We show the perceptual quality of the reconstructed image (LPIPS defined in Eq. 3.2) at different levels of noise perturbations (measured using signal-to-noise ratio). For OOD images such as CXR, the resulting solution ($(\mathcal{W}+, \mathcal{S}, \mathcal{B})$ in this illustration) is highly non-robust that even a minor perturbation introduces face-like features into the reconstruction. In contrast, SPHInX produces a solution that is perceptually more accurate as well as robust under perturbations.

solutions that are more locally robust for OOD images, so that the algorithm can stably converge even for severely ill-posed problems, e.g., compressed sensing.

3.2.2 Optimization with Projection Heads

We propose to improve the StyleGAN inversion process by optimizing a projection head that maps between $\mathcal{Z}+$ and $\mathcal{W}+$, instead of directly searching in either of the latent spaces via gradient descent. Intuitively, this crucial modification requires the inversion technique to transform the prior distribution $P(\mathcal{Z}+)$ into an appropriate latent distribution $P(\mathcal{W}+)$, such that any realization from $P(\mathcal{W}+)$, when passed to the StyleGAN generator $\mathcal{G}$ will reconstruct the given image I. For example, in the style latent space, the projection head $\mathcal{P}_s(.)$ takes a realization from $P(\mathcal{Z}+)$, $\mathbf{z}^+ \in \mathcal{Z}+ \subseteq \mathbb{R}^{N_\ell \times 512}$ to produce a projected latent code $\mathbf{w}^+ \in \mathcal{W}+ \subseteq \mathbb{R}^{N_\ell \times 512}$.

A naïve way to implement this is to directly update the pre-trained mapping function f to perform OOD inversion, without directly manipulating the latent variables as done in all existing approaches. However, as showed in Figure 11, this results in poor quality embeddings and reconstructions $\hat{I}$ that do not contain any of the characteristics in the input image. One potential explanation for this behavior is that the pre-trained f contains strong inductive biases for recovering face images and it is non-trivial to adapt that function to severe OOD data. To validate this hypothesis, we randomly re-initialized f and repeated the inversion experiment. Surprisingly, this improves the optimization process and produces images that somewhat resemble patterns found in an X-ray. However, we find that the quality of the reconstruction is still far from being optimal. This can be attributed to the fact that, even with in-distribution images, the inversion can be improved by individually controlling every latent vector in $\mathbf{w}^+$ (Richardson et al. 2021). Alternately, ILO (Daras et al. 2021) used a fixed mapping f, but adopted a novel optimization strategy that progressively included latent variables from different layers of StyleGAN and optimized for latent

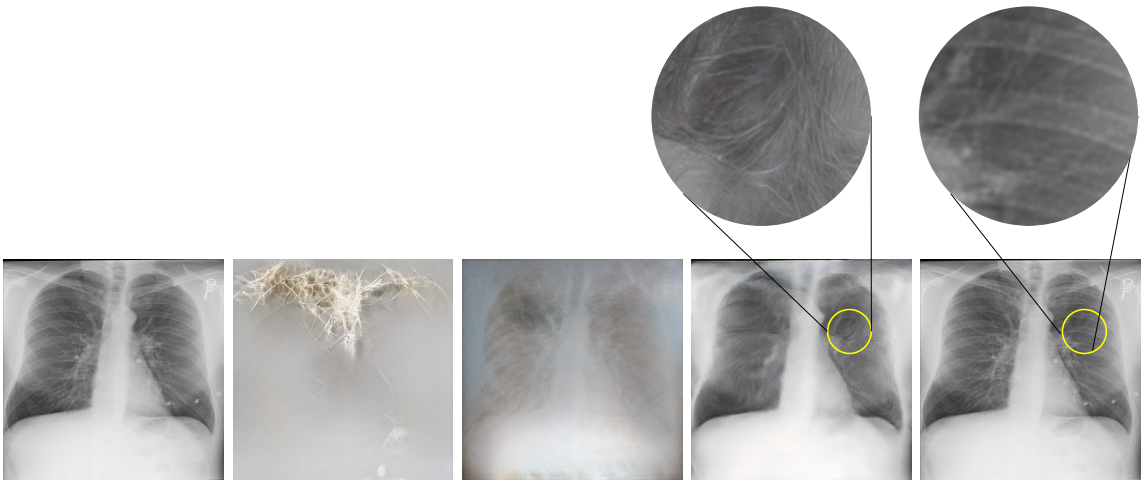


Figure 11. **Design of the projection head**. While re-purposing the pre-trained mapping function f from StyleGAN-v2 as the projection head fails completely, randomly re-initializing f produces reasonable images. However, using the proposed projection head $\mathcal{P}_s$, which decouples the different latent spaces in $\mathcal{W}+$, leads to significantly higher quality reconstructions.

codes that lie within an ℓ_1-ball around the manifold induced by the previous layer. However, the inherent lack of local robustness for OOD images in the $\mathcal{W}+$ latent space makes such a progressive optimization also challenging. For example, as showed in Figure 11, though the CXR reconstruction obtained using ILO looks significantly better than using a randomly initialized mapping function f, from a closer look, we notice that the image contains hair-like patterns instead of bone structure, indicating the leakage of face-specific characteristics into OOD images.

To circumvent this challenge, we design the *style projection head* $\mathcal{P}_s$ such that it decouples the latent spaces for different layers in $\mathcal{W}+$. In other words, $\mathcal{P}_s$ transforms each $\mathbf{z}^+ \in \mathcal{Z}+$ into $d-$dimensional representations using a *bottleneck* block of MLP layers. Subsequently, N_ℓ different decoder blocks (again a set of MLP layers) independently provide the corresponding mapping $\mathbf{w}^+ \in \mathcal{W}+$, using the bottleneck

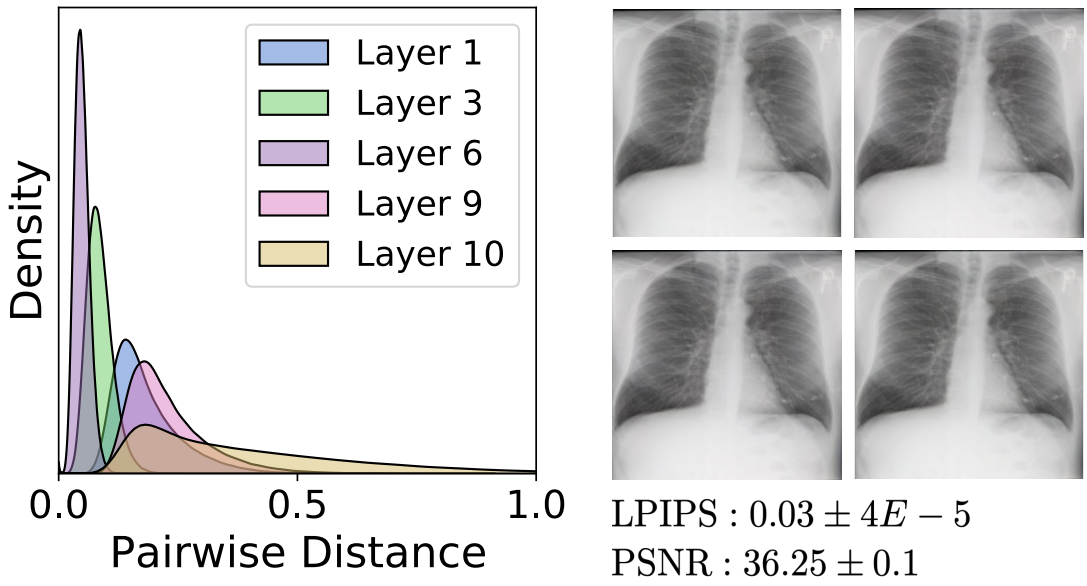


Figure 12. **Behavior of the projection head**. We demonstrate our ability to approximate $P(\mathcal{W}+)$ by generating 1000 realizations of $\mathcal{P}_c(\mathbf{z}^+)$ and visualizing the distribution of pairwise distances between the resulting $\mathbf{w}'$s in each layer. Interestingly, the reconstructions corresponding to the 1000 realizations are perceptually very similar, thus indicating a stable convergence of SPHInX.

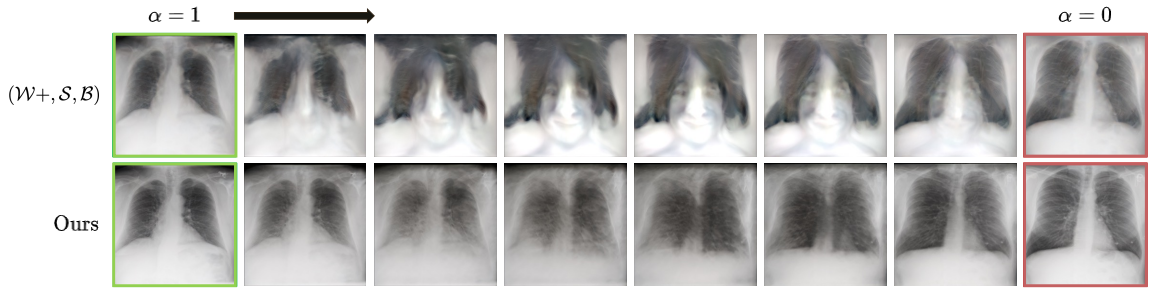


Figure 13. **Interpolating in the StyleGAN-v2 latent space**. Similar to our observation in Figure 10, a latent walk between two different X-ray images produces face images, when a conventional GAN inversion method is adopted. On the other hand, SPHInX produces highly plausible realizations as we transition between the the two inputs.

representation as input. Note that, while each $\mathbf{z}^+ \in \mathbb{R}^{512}$ and $\mathbf{w}^+ \in \mathbb{R}^{512}$, the choice of bottleneck dimension d is not very sensitive and we used $d = 16$ in all our experiments. Interestingly, using the proposed projection head (randomly initialized) and optimizing

with different realizations of $\mathbf{z}^+$ from $P(\mathcal{Z}+)$ in every iteration of the optimization process, we are able to obtain an accurate estimate of $P(\mathcal{W})+$ for a given image. As illustrated in Figure 11, this results in a high-fidelity reconstruction of even OOD images using a pre-trained StyleGAN-v2.

In addition to the input latent space, StyleGAN also uses a content latent space $\mathbf{s} \in \mathcal{S} \subseteq \mathbb{R}^{4\times4\times512}$ to represent the semantic structure (referred to as content) of the in-domain images, and is optimized during the training of the generator. When inverting an image, which can potentially contain unrelated structure, the direct adoption of the pre-trained content input from StyleGAN can be ineffective. Hence, it is common to include $\mathcal{S}$ to the collection of latent variables that we optimize (Y. Wang et al. 2020; Kang, Kim, and Cho 2021). Based on our intuition about utilizing projection heads for OOD image inversion, we also introduce a content projection head $\mathcal{P}_c$ to directly parameterize the content input (randomly initialized using a Gaussian prior in lieu of the pre-trained content tensor). Finally, following the findings from ILO (Daras et al. 2021), we also include single channel Gaussian noise inputs ($\mathcal{B}$) from StyleGAN-v2, corresponding to each layer of the synthesis network, in our optimization to improve the perceptual quality of the reconstructed images.

3.2.3 Observations

In contrast to existing approaches (Abdal, Qin, and Wonka 2019; Daras et al. 2021), SPHInX replaces the mapping function f in StyleGAN-v2 using a projection head $\mathcal{P}_s : \mathcal{Z}+ \rightarrow \mathcal{W}+$, which is optimized for different realizations of $\mathbf{z}^+ \sim P(\mathcal{Z}+)$ in every iteration. Using such a strategy, we are able to obtain an accurate and a robust estimate of $P(\mathcal{W}+)$ for a given image. To demonstrate this, after SPHInX

training, we generated $1,000$ realizations of $\{\mathcal{P}_c(\mathbf{z}^+), \forall \mathbf{z}^+ \sim P(\mathcal{Z}+)\}$ and visualize the distribution of pairwise distances between the resulting $\mathbf{w}'$s in each layer in Figure 12. While the varying widths of the densities indicate that there are different levels of sensitivity in each of the layers, the reconstructions corresponding to all 1000 cases represents minimal deviation in their perceptual quality. This clearly evidences the stable convergence of SPHInX to an accurate solution even for OOD images. Consequently, as showed in Figure 10, the embeddings from SPHInX are more robust to local perturbations. Finally, we try to interpolate between two images I_1 and I_2 in the StyleGAN latent space using their corresponding embeddings $\mathbf{w}_1^+$ and $\mathbf{w}_2^+$, *i.e.*, $\mathbf{w}_*^+ = \alpha \mathbf{w}_1^+ + (1-\alpha)\mathbf{w}_2^+$, where $\alpha \in [0, 1]$. In comparison with the $(\mathcal{W}+, \mathcal{S}, \mathcal{B})$ baseline, as illustrated in Figure 13, we find that our proposed approach produces a meaningful transition without introducing any in-domain characteristics.

3.3 Experiment Setup

3.4 Out-of-Distribution Image Reconstruction

Corroborating the observations from Figure 9, Figures 14 - 14 illustrate the out-of-distribution image reconstruction performance of SPHInX over the baselines. For each dataset, we show the median, along with the 25^{th} and the 75^{th} percentiles, of the metrics obtained from 50 randomly chosen images. We find that SPHInX consistently outperforms the baselines across all datasets, thereby demonstrating its efficacy in OOD inversion. Interestingly, the performance of state-of-the-art approaches such as I2S++ (referred as $(\mathcal{W}+, \mathcal{S})$) and ILO (equivalent to $(\mathcal{Z}+, \mathcal{S}, \mathcal{B})$ when the ℓ_1-ball constraint is removed) are very similar across all metrics, and consistently lower

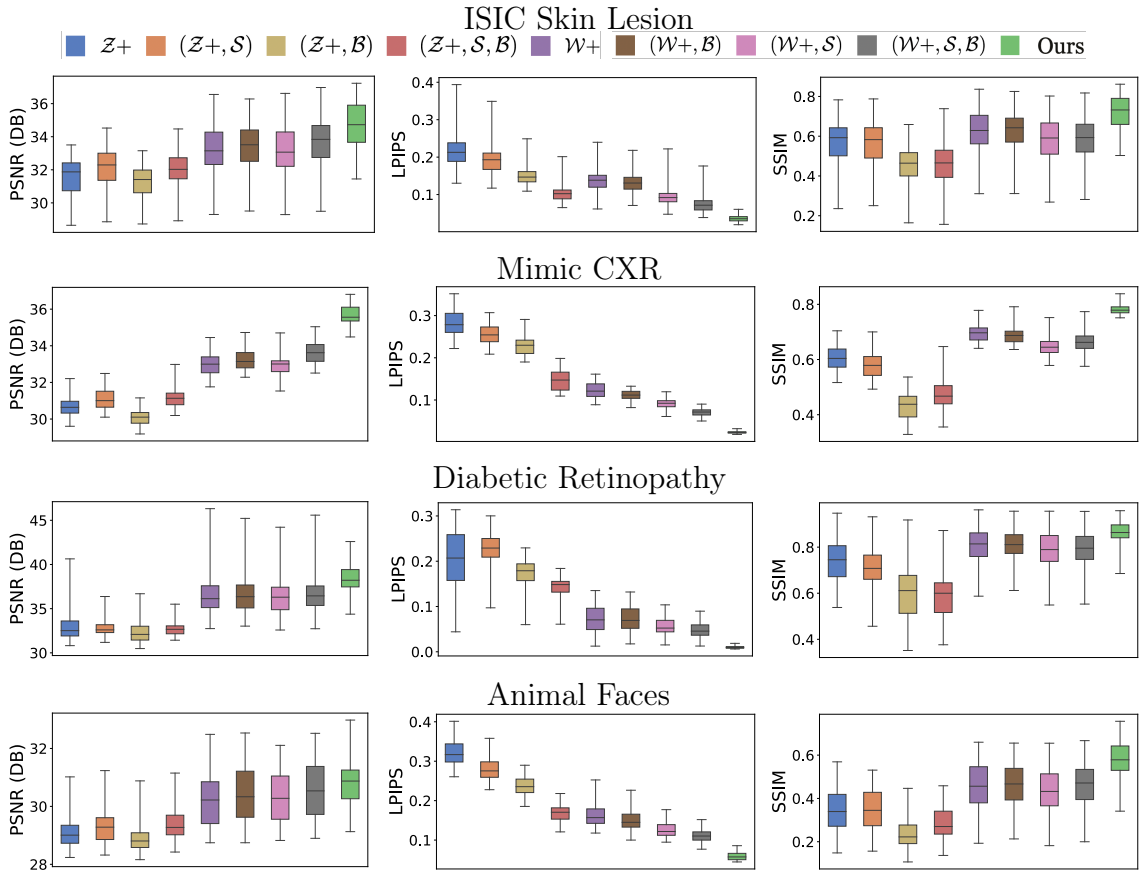


Figure 14. **Comparison of OOD image reconstruction performance**. Through the use of style and content projection heads, along with a novel training strategy, we find that SPHInX consistently outperforms the baseline methods in all the metrics (LPIPS↓, PSNR↑ and SSIM↑) across the datasets.

than SPHInX. Comparatively, the $(\mathcal{W}+,\mathcal{S},\mathcal{B})$ baseline (not reported so far in the literature) is the second best approach for OOD inversion.

3.5 Ill-posed Image Restoration

Next, we evaluate SPHInX on a suite of ill-posed image restoration tasks (Daras et al. 2021). For all tasks considered, we utilized a deterministic corruption function that operates upon the true image I to produce the observation I'. It must be noted

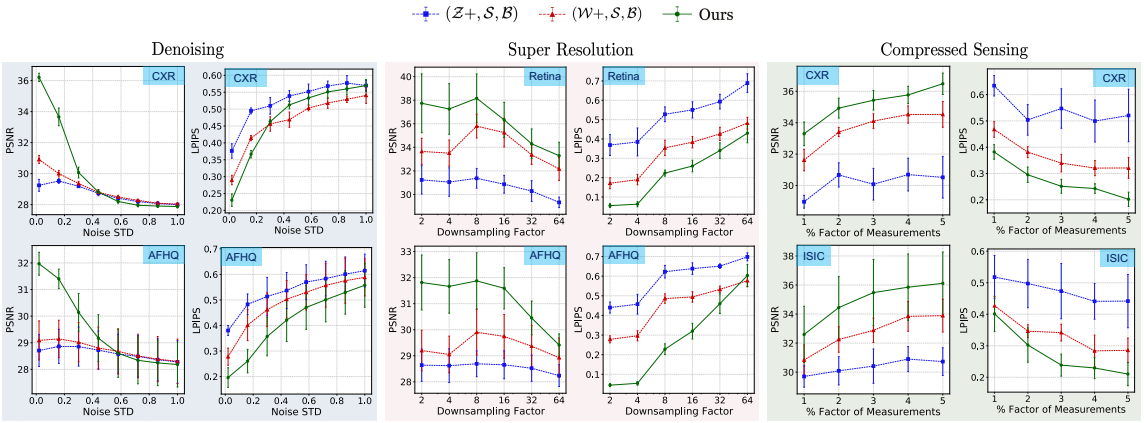


Figure 15. **Comparison of SPHInX against different baselines in ill-posed image restoration**. Even in the absence of image-specific priors, we observe that SPHInX effectively recovers the true image, as evidenced by the improvement in the PSNR↑ and LPIPS↓ metrics.

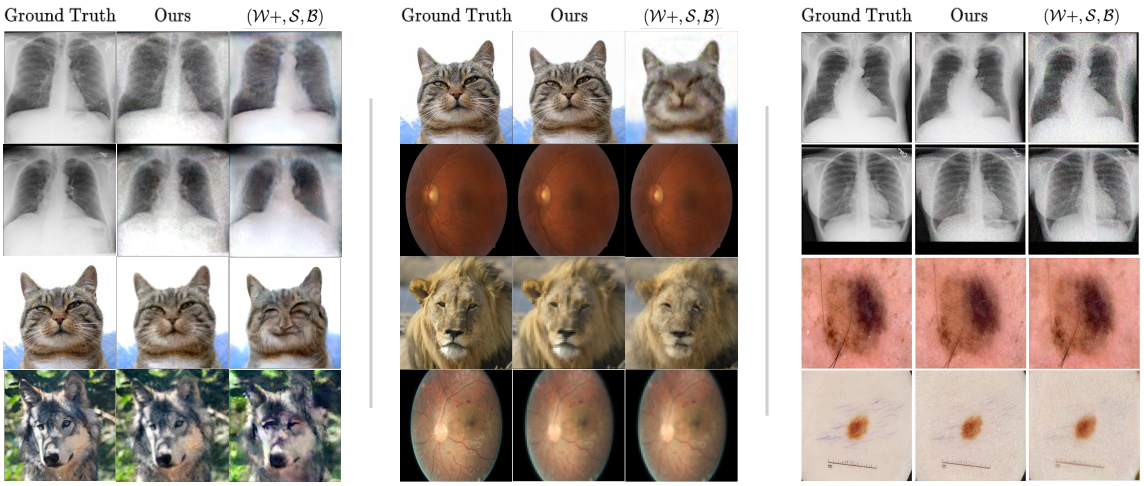


Figure 16. **Reconstructed images from ill-posed inversion**. (left-right) denoising at noise std = 0.3, super resolution at downsampling factor = 8 (rows 1, 2), and 16 (rows 3, 4), and compressed sensing at 1% measurements. In each we show the ground truth image along with the reconstructions from the $(\mathcal{W}+,\mathcal{S},\mathcal{B})$ baseline and SPHInX.

that the inverse optimization is only exposed to the observation I', while the true image I is only used for evaluation.

Denoising. Our goal is to recover a clean image given its corrupted observation. We added known Gaussian noise to every image I to synthesize I' and clipped the observed

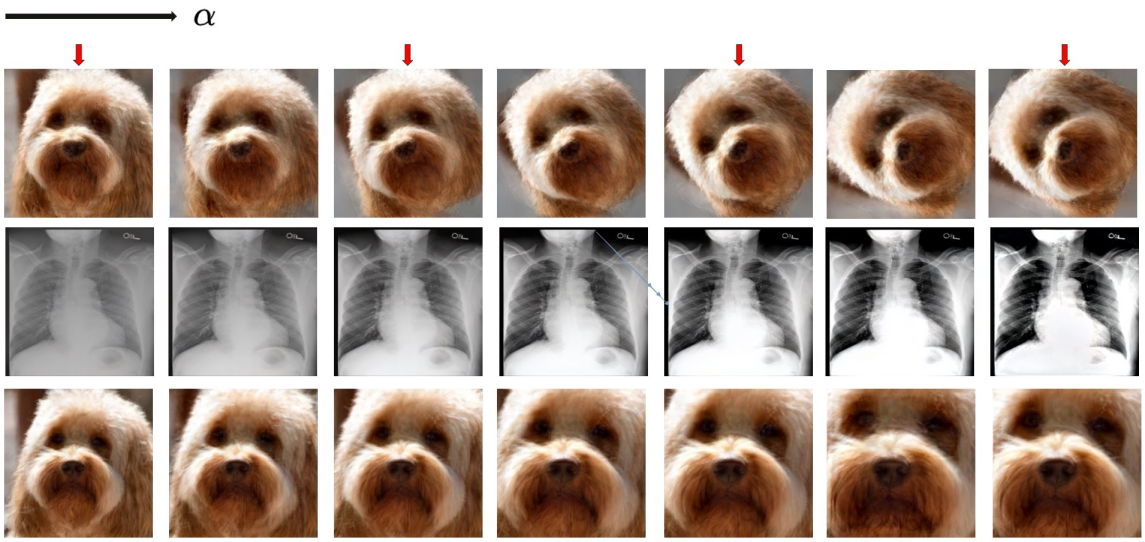


Figure 17. **Simultaneous image inversion and attribute discovery**. SPHInX can learn meaningful attribute directions - rotation (first row), brightness (second row) and zoom (last row) - by simultaneously inverting an image along with its realizations that differ by the attribute. By varying the scale of traversal α along the inferred direction, we observe that SPHInX effectively produces realizations reflective of the learned attribute. In each case, the input images are marked with a red arrow.

image to the range $[-1, 1]$. The loss function is a combination of mean-squared error (MSE) and LPIPS between the generated and true observations ($\mathcal{G}(\mathbf{w}^+)$ and I').

Super Resolution. This refers to the ill-posed problem of increasing the resolution of an observed low-resolution. For this task, we synthesized I' by downsampling (bi-cubic interpolation (Daras et al. 2021)) the true image I with a known factor. Note, we perform the same downsampling operation to the generated image from StyleGAN-v2 before loss computation. For this optimization, we used both MSE and LPIPS losses. We did not include the geodesic loss contrary toMenon et al. 2020; Daras et al. 2021 as it provided poor quality results for OOD images.

Compressed Sensing. This task seeks to reconstruct an image from a small number of linear and non-adaptive measurements. For this task, following (Daras et al. 2021),

we generated observations of random projections using partial circulant measurement matrices with random signs and a known percentage of measurements. It must be noted that, we use the same measurement process, as that of the true image, with the StyleGAN-v2 reconstruction before evaluating the loss function. For the optimization, we minimized the MSE loss between the generated and the true observations.

Findings. Figure 16 provides a visual comparison of the restored images using SPHInX and the best-performing baseline ($\mathcal{W}+, \mathcal{S}, \mathcal{B}$) across the three restoration tasks. Figure 15 reports the metrics aggregated across 10 different examples and varying levels of corruption severity. More specifically, we show the mean and standard deviation for each of the metrics in each case. For the task of denoising, we measured the performance over increasing noise strengths (std) in the range $[0.20, 1]$. With super-resolution, we measured performance over a range of downsampling $[2, 128]$, where a factor of 128 implies that the observed image will be of size 8×8. Finally, for compressed sensing, we varied the factor (%) of measurements in the range of $[1, 5]$. It can be observed that SPHInX consistently outperforms the baselines (higher PSNR and lower LPIPS) under most of the restoration tasks. However for the task of denoising, under very high noise strengths, SPHInX exhibits a behavior similar to the other baseline methods.

3.6 Simultaneous Image Inversion with Attribute Discovery

A desired property of any GAN inversion algorithm is that the latent codes can be semantically manipulated for downstream applications such as style transfer and attribute discovery Voynov and Babenko 2020; Härkönen et al. 2020a; Plumerault, Borgne, and Hudelot 2020; Jahanian, Chai, and Isola 2019; T. Wang et al. 2021.

However, when inverting OOD images onto the GAN latent space, the mismatch between the latent space prior and the OOD image makes it significantly hard to meaningfully manipulate the latent codes. Hence, we introduce a new inverse optimization problem to evaluate GAN inversion techniques. Given an image I along with its K variants that differ by a single attribute (for e.g., rotation), our goal is to simultaneously invert all $K+1$ images using a starting point $\mathbf{w}_*^+$ in the latent space for embedding I and local direction vectors in each of the layers, $\mathbf{D} \in \mathbb{R}^{N_\ell \times 512}$, along which the remaining K variants can be accurately embedded. Formally,

$$\min_{\mathbf{w}_*^+, \mathbf{D}, \{\alpha\}} \mathcal{L}(\mathcal{G}(\mathbf{w}_*^+), I) + \sum_{k=1}^{K} \mathcal{L}\left(\mathcal{G}\left(\mathbf{w}_*^+ + \alpha_k \frac{\mathbf{D}}{\|\mathbf{D}\|_2}\right), I_k\right),$$

where $\{\alpha_k\}$ refers to the set of scaling parameters for each of the K images. Upon training, we expect the generator to synthesize manipulations pertinent to the learned attribute by traversing along $\mathbf{D}$ from $\mathbf{w}_*^+$.

Setup. In this study, we considered three different image transformations: (i) rotation, (ii) brightness and (ii) zoom and synthesized $K = 3$ different variants for each image by manipulating the chosen attribute.

Findings. Figure 17 illustrates the results from SPHInX corresponding to all three attributes. The images are generated by traversing the learned direction vector $\mathbf{D}$ by varying α. Our results show that, SPHInX can accurately recover directions corresponding to specific attribute changes in the StyleGAN latent space. This experiment clearly establishes SPHInX as a powerful GAN inversion method for challenging inverse problems with OOD data.

3.7 Conclusions

In this chapter we presented SPHInX, a new approach for solving ill-posed inverse problems with pre-trained StyleGAN-v2. Through the use of carefully designed projection heads for style and content latent spaces, and a novel training strategy, SPHInX produces accurate and robust embeddings for even arbitrary OOD images. With extensive empirical studies with multiple datasets, we demonstrated significant performance improvements in embeddings high-resolution OOD images as well as ill-posed tasks such as denoising, super resolution and compressed sensing. Compared to state-of-the-art approaches such as I2S++ (Abdal, Qin, and Wonka 2020) and ILO (Daras et al. 2021), we find that SPHInX stably converges to meaningful embeddings in the latent space and effectively avoids leakage of face-specific characteristics into the reconstructions. In summary, our study clearly evidences the utility of StyleGAN as a strong image prior even in domains where collecting large datasets for training custom generative models is infeasible.

Chapter 4

EXPLORING THE UTILITY OF CLIP PRIORS FOR VISUAL RELATIONSHIP PREDICTION

In this chapter, we study the problem of accurately estimating the relationship between objects in a scene. While it is common to leverage image features (e.g., extracted using an object detector), state-of-the-art visual relationship prediction (VRP) methods adopt sophisticated graphical models, comprising both image features and text descriptors (D. Xu et al. 2017; Zellers et al. 2018; J. Yang et al. 2018; K. Tang et al. 2019). In this context, one might expect multimodal joint embeddings such as CLIP (Radford et al. 2021a) to be beneficial. However, recent research (Zhao et al. 2022) finds the language priors from CLIP to be limiting for practical reasoning tasks. We make a similar observation about VRP that CLIP's text embeddings are insufficient to distinguish between different predicates relating a subject-object pair, e.g., *horse-grass*. In Figure 19, we visualize the cosine similarities between the CLIP embedding of the query image and CLIP embeddings of text descriptors constructed using different predicate choices, *i.e.*, <horse, predicate, grass>. Interestingly, CLIP suggests *in front of*, *growing on* and *eating* as plausible predicate choices. This naturally raises the question: can we leverage CLIP for VRP, and if so, how?

To this end, we introduce CREPE (**C**LIP **R**epresentation **E**nhanced **P**redicate **E**stimation), which utilizes learnable prompts and a novel contrastive training strategy to infer reliable CLIP representations for union boxes (obtained by combining *subject* and *object* bounding boxes). Note, CREPE can be used with any existing VRP method

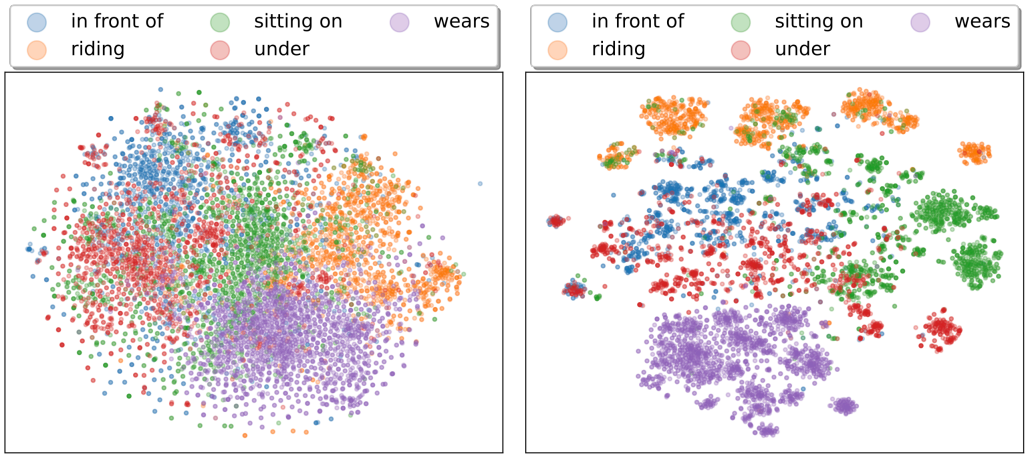


Figure 18. T-SNE visualization of the predicate representations from UVTransE trained with: (left) CLIP-based image embeddings for subject, object and union box regions; (right) CLIP-based image embedding for the union box, along with CLIP-based text embeddings for subject and object boxes.

(e.g., UVTransE, VCTree) and in practice, leads to superior generalization. We find, with these union box representations, one can effectively infer visual predicates even using naïve CLIP text embeddings for the subject and object boxes (*i.e.*, object labels).

Through rigorous experimentation on the *Visual Genome* (*VG*) benchmark, we show that incorporating `CREPE` into the vanilla implementations of UVTransE and VCTree improves their performance significantly, without needing additional training strategies or calibration techniques. Remarkably, `CREPE` achieves state-of-the-art **mr@**K performance in the challenging setting of low values of K with minimal degradation in performance. For example, with UVTransE, `CREPE` achieves **mR@**$5 = 27.79$ and **mR@**$20 = 31.95$. Finally, `CREPE` also reveals strong generalization to diverse and previously unseen predicate occurrences from the *Unrel* benchmark, despite lacking explicit training on such examples.

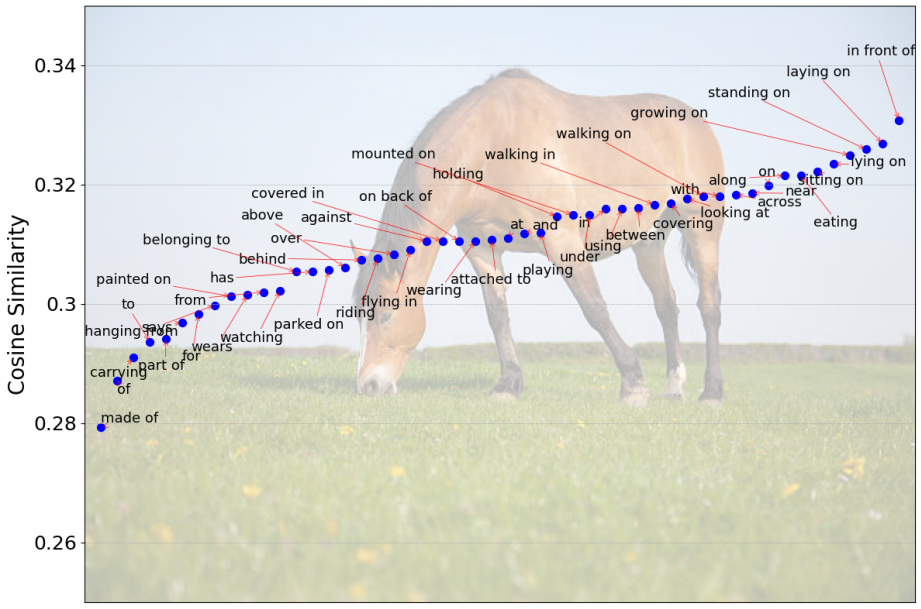


Figure 19. **CLIP embeddings are insufficient to distinguish between different predicate choices in VRP.** We plot the cosine similarity between CLIP embedding of the query image and CLIP text embeddings for different predicates (<horse, predicate, grass>).

4.1 Related work

Visual relationship prediction is central for scene graph generation and to enable situational awareness in navigation and surveillance tasks. A large class of VRP methods can be found in the literature, and they differ in terms of the model used for representing the triplet <subject, predicate, object> and the training strategy adopted (e.g., loss function, handling long tail bias etc.). Popular examples include IMP (D. Xu et al. 2017) that uses a graph neural network to capture higher-order dependencies and contextual information, UVTransE that adopts a translational embedding model, MOTIFS (Zellers et al. 2018) that employs a frequency-based calibration technique, Graph RCNN (J. Yang et al. 2018) that uses a novel architecture to integrate object detection and relationship classification, and VCTree (K. Tang

et al. 2019) that uses dynamic tree structures to capture the local and global visual contexts.

While both vision and language features are exploited by the aforementioned methods, more recently, the use of vision-language models (VLM) in VRP has also been explored. For example, Yu *et al.* (Q. Yu et al. 2023) utilize image-caption pairs from external datasets (COCO, CC3M) along with a pre-trained VLM to augment the *Visual Genome* dataset with fine-grained predicates. On the other hand, RelCLIP (Y. Zhu et al. 2022) leverages a pre-trained CLIP model to decouple each triplet into subject-predicate and object-predicate embeddings, thereby improving the alignment between a predicate and the visual features. In contrast, `CREPE` performs image-conditioned prompting with CLIP to obtain more reliable representations for union boxes, while not requiring any additional data. Consequently, `CREPE` is flexible enough to be incorporated into any existing VRP method.

4.2 Proposed Approach

The goal of visual relationship prediction is to predict the interaction between a pair of objects in an image. Such relationships can be spatial, action-based, semantic, or comparative in nature and are commonly represented as triplets in the form of <`subject`, `predicate`, `object`>, for example <`person`, `riding`, `bike`>. Our key objective is to leverage the inherent priors in CLIP to improve the performance of any VRP method.

Most existing VRP methods require feature representations for the subject and object bounding boxes as well as the union box obtained by combining the two. While it is straightforward to obtain image-based (e.g., latent features from an object

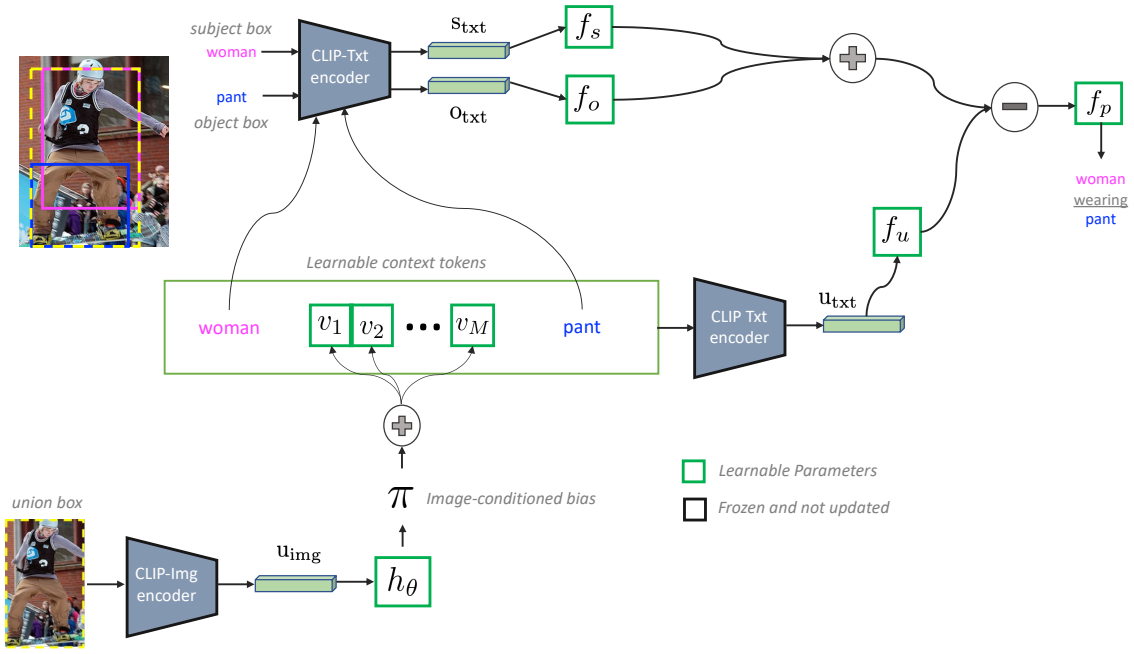


Figure 20. **Incorporating `CREPE` into UVTransE**. An illustration of how UVTransE can be implemented with `CREPE` training. `CREPE` uses learnable context vectors along with image-conditioned bias correction to obtain visually grounded text descriptors for an union image. Note, the CLIP backbone is used to both perform the optimization for text prompt generation as well as producing the embeddings (`s_txt`, `o_txt`, `u_img`).

detector), text-based (e.g., text embeddings for the object labels) or even hybrid (*i.e.*, text + image) representations for the subject and object regions, constructing reliable descriptors for the union box is challenging. Our work focuses on leveraging CLIP to construct these representations, with the goal of improving the robustness of VRP methods.

Union Bounding Box Representations. At the outset, a straightforward approach is to compute CLIP image embeddings for the union boxes; however we find that it provides no apparent benefits over image features from a pre-trained object detector. Hence, an alternative approach is to construct or even automatically learn a meaningful text prompt for the union box (Liu and Chilton 2022). Even

with the powerful visual grounding that CLIP enables, such prompts are not guaranteed to sufficiently distinguish between different predicates (see Figure 19). To address this, CREPE proposes to construct prompts for union images (in the form <subject, learnable − text − tokens, object>) through a novel contrastive training strategy, which jointly leverages CLIP priors and also promotes maximal separation between predicates.

The learnable text tokens $[\mathrm{v}_1, \cdots, \mathrm{v}_M]$ are randomly initialized in the CLIP's token embedding space, with these M vectors being updatable during training. Using these tokens, we construct the text prompt for a union image as $\{\mathrm{s}_t - \{\mathrm{v}_i\}_{i=1}^{M} - \mathrm{o}_t\}$, where s_t and o_t are the subject and object token embeddings. While these context tokens are the same for all cases, we adopt an approach similar to recent methods (Zhou et al. 2022; Khattak et al. 2023) by incorporating image-conditioned biases during prompt learning. More specifically, we construct a non-linear MLP $h_\theta(.)$ that takes the CLIP image embedding for a union bounding box as input and outputs the bias $\text{invalid} = h_\theta(\mathrm{u}_{\text{img}})$. In other words, each of the M learnable text tokens are refined with the image-conditioned bias as $\mathrm{v}_m = \mathrm{v}_m + \text{invalid}$. We refer to the prompt generation process as $g_\phi(.)$, where the learnable parameters ϕ correspond to the context tokens $\{\mathrm{v}_m\}$ and parameters θ of the MLP h_θ.

CREPE training involves a cross-modal retrieval strategy, where we first generate a pseudo label for each union image, and subsequently leverage that as the negative sample for the constrastive objective. Specifically, we construct a vocabulary of all potential <subject, predicate, object> triplets in the visual genome (VG) benchmark and select the most similar triplet for each union image based on cosine similarity of CLIP embeddings. Formally, the CLIP image embedding of the union bounding box is denoted as u_{img}, and the CLIP embedding for the learned text prompt as

Model	mR@5	mR@10	mR@15	mR@20	mR@50
MOTIFS [2]	-.-	-.-	-.-	11.5	14.6
+ TDE [11]	-.-	-.-	-.-	18.5	25.5
SG-Transformer [12]	-.-	-.-	-.-	14.8	19.2
+ CogTree [12]	-.-	-.-	-.-	22.9	28.4
+ SCR [13]	-.-	-.-	-.-	27.0	32.2
PE-Net (P) [14]	-.-	-.-	-.-	-.-	23.1
PE-Net [14]	-.-	-.-	-.-	-.-	31.5
VCTree [4]	-.-	-.-	-.-	11.7	14.9
+ PPDL [15]	-.-	-.-	-.-	-.-	33.3
+ SCR [13]	-.-	-.-	-.-	27.7	33.5
+CREPE	25.93	29.88	30.67	30.91	31.01
UVTransE (visual only) [16]	-.-	-.-	-.-	8.26	11.41
+ visual + language	-.-	-.-	-.-	14.33	19.50
+CREPE	27.79	31.12	31.78	31.95	32.09

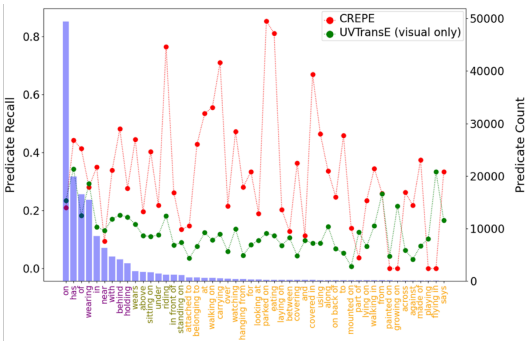


Figure 21. **Performance evaluation of CREPE.** (left) We study the utility of CREPE with two popular VRP methods, UVTransE and VCTree, on the Visual Genome (VG) benchmark using the mean Recall@K (mR@K) metric. The best performing method is highlighted in red, while the second best is in blue; (right) We show the R@50 performance for each of the predicate classes obtained using UVTransE, along with the frequency of occurrence. The recall values are shown as dotted lines, while the frequencies are displayed as blue bars.

$\mathrm{u}_{\mathrm{txt}} = \mathrm{CLIP}_{\mathrm{txt}}\big(g_\phi(\mathrm{s}_{\mathrm{txt}}, \mathrm{o}_{\mathrm{txt}}, \mathrm{u}_{\mathrm{img}})\big)$. We adopt the following contrastive objective to infer the parameters ϕ:

$$\mathcal{L} = -\log \frac{\exp\big(\mathrm{sim}(\mathrm{u}_{\mathrm{img}}, \mathrm{u}_{\mathrm{txt}})\big)}{\exp\big(\mathrm{sim}(\mathrm{u}_{\mathrm{img}}, \mathrm{u}_{\mathrm{txt}})\big) + \exp\big(\mathrm{sim}(\mathrm{u}_{\mathrm{img}}, \hat{\mathrm{u}}_{\mathrm{txt}})\big)}. \tag{4.1}$$

Here, sim denotes the cosine similarity and $\hat{\mathrm{u}}_{\mathrm{txt}}$ is the CLIP-text representation for the pseudo labels obtained via cross-modal retrieval with the VG triplet vocabulary. Intuitively, this optimization encourages text prompts that can more appropriately describe the content of the union image compared to its pseudo-label. Upon completion of this contrastive training process $g_\phi(.)$, we incorporate the CLIP-text embedding for the object labels of subject and object, and that of the union box prompt into any VRP method (e.g., UVTransE, VCTree).

4.3 Experiments

4.3.1 Setup

We evaluate our approach on Visual Genome (VG) (Krishna et al. 2017), which has 108,077 annotated images with 150 unique object categories and 50 unique predicates. Our train, test and val splits had 57,772, 26440 and 5000 images. For evaluation, we predict relations between entities using ground truth bounding boxes and labels. We employ the Mean Recall@K (mR@K) metric, specifically focusing on smaller K values for better real-world relevance, and report results for $K = [5, 10, 15, 20, 50]$.

Implementation Details: `CREPE`'s $h_\theta(.)$ employs a 2-layer MLP: FCN -> ReLU -> FCN. $g_\phi(.)$ training uses SGD with LR 2e$-$3 for 500 epochs. Following conventional practice, we train our models with a *no-relation* class but exclude it in the recall metrics computation. The baseline models uses frozen Faster R-CNN (Ren et al. 2015) for visual embeddings, following established practices (K. Tang et al. 2019; K. Tang et al. 2020; Zellers et al. 2018; Tianshui Chen et al. 2019). In our implementation, we employ the CLIP ViT-B/32 model, which remains frozen throughout our process.

4.3.2 Results

We implemented `CREPE` with UVTransE and VCTree, two prominent methods in visual relationship prediction, and present a comparative analysis in Figure 21. It is worth emphasizing that our work is the first to report mR@5, 10, 15, and therefore, the results of other methods at these values are not available. In Figure 21,

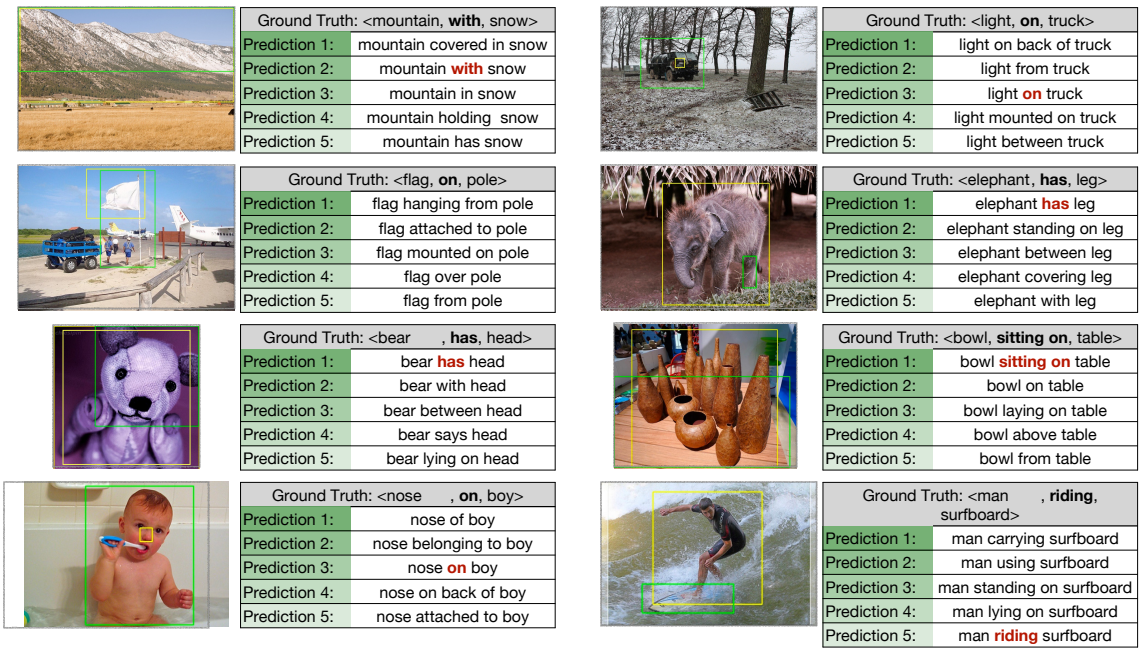


Figure 22. **Qualitative Results from UVTransE with** `CREPE`. Each sub-figure shows the relationship between a subject (yellow box) and an object (green box), accompanied by the top five predictions. Correct predictions are highlighted.

UVTransE (visual only) utilized image only representations from the object detector, while UVTransE (visual & language) combines image and text embeddings.

Key Findings: `CREPE` leads to significant performance gains over the vanilla implementations of UVTransE and VCTree methods. In particular, `CREPE` with UVTransE achieves an mR@20 score of 31.95, which is an improvement of 4.25 percentage points (or 15.3% relative gain) compared to the second-best performing method, VCTree with SCR bias correction. More impressively, mR@5 performance of our method at 27.79 surpasses even the previously best mR@20 performance (via VCTree) as shown in Figure 21 (left). Hence, there is no degradation in quality at much smaller recall thresholds, underscoring the effectiveness of our approach. Figure 21 (right), shows that `CREPE` exhibits superior performance even in the tail predicates, when compared

to the vanilla UVTransE implementation. This highlights how CREPE representations lead to improved generalization behavior.

Figure 22 demonstrates the quality of predictions with CREPE. For instance, with the first example, for the ground truth <mountain, with, snow>, the top prediction from UVTransE + CREPE is <mountain, covered in, snow>. However, we note that it matches the ground truth with its second prediction. Other examples similarly demonstrate CREPE's ability to handle diverse types of images and relations.

Figure 23. **CREPE improves robustness of VRP.** Here, we illustrate the results on the Unrel dataset, which contained previously unseen objects and atypical relationships.

CREPE Enhances VRP Robustness: To assess the generalizability and robustness of CREPE's representations, we conduct evaluations on the Unrel dataset (Peyre et al. 2017). In this experiment, we directly apply UVTransE + CREPE trained on VG using CREPE's representations. The Unrel dataset presents a unique challenge, as it includes atypical relations not found in the VG benchmark, such as

Table 12. **Ablations.** This table compares the performance of UVTransE with CREPE representations. We show variants where we do not include learnable text prompts but directly utilize pseudo labels from Cross-Modal Retrieval for union box representation. The best performing method is highlighted in red, while the second best is in blue.

Model	mR@5	mR@10	mR@15	mR@20	mR@50
UVTransE (visual only) (Hung, Mallya, and Lazebnik 2020)	3.93	6.39	7.76	8.26	11.41
+ language	6.33	9.85	12.30	14.33	19.50
UVTransE + CREPE **(w/o learnable prompting)**					
+ Pseudo Labels ($K = 1$)	8.49	12.45	15.15	17.13	22.53
+ Pseudo Labels ($K = 3$)	9.85	14.40	17.36	19.48	24.61
+ Pseudo Labels ($K = 4$)	9.88	14.42	17.25	19.12	24.86
+ Pseudo Labels ($K = 5$)	9.97	13.84	17.04	19.22	24.60
UVTransE + CREPE	27.79	31.12	31.78	31.95	32.09

the <car, on the top of, bus> triplet. Despite this shift, CREPE demonstrates its effectiveness in estimating these unconventional relations, even without specific training on such examples, as illustrated in Figure23. For example, in the case of <elephant, cover, car>, CREPE's top prediction of <elephant, standing on, car> accurately reflects the scene. Furthermore, in the challenging instance of <dog, wearing, sunglasses>, where both the relationship and the object *sunglasses* are atypical in VG, CREPE still performs well.

Ablations: We conducted an ablation study to assess the significance of learnable context tokens in generating a text based representation for the union image. For this study, we utilized pseudo labels obtained from Cross-Modal Retrieval (CMR), where the labels are retrieved from an exhaustive vocabulary of all possible <subject, predicate, object> triplets in the Visual Genome (VG) benchmark.

We explored two scenarios: 1) Selecting the most similar entry from the triplet vocabulary for each union image based on cosine similarity of CLIP embeddings. 2) Selecting the top K similar entries with an attention module (Ilse, Tomczak, and Welling 2018) to capture context. Results in Table 12 demonstrate the effectiveness of pseudo labels, notably outperforming CLIP-based, visual-only, and visual+language

UVTransE baselines. Using context ($K > 1$) consistently improved predicate estimation, yielding a 2.5% mR@50 gain compared to $K = 1$. However, learnable text tokens still outperformed pseudo labels, showcasing their superiority in complex relationship representation beyond a restricted vocabulary used for triplet selection.

4.4 Conclusion

In summary, our study unveils the untapped capabilities of vision-language models, particularly CLIP, in visual relationship prediction. Our method `CREPE` leverages text-based representations and a unique contrastive training strategy to infer reliable union box representations for improving VRP, while simultaneously tackling the long-tail issue prevalent in predicate occurrence distribution. The implications of our work are broad, with potential advancements in numerous applications like autonomous navigation and intelligent surveillance.

Chapter 5

PRIME: LEVERAGING VISION-LANGUAGE PRIORS FOR IMPROVED MODEL FAILURE DETECTION AND EXPLANATION

With the increasing adoption of deep models in diverse applications (Chib and Singh 2023), ensuring performance reliability and limiting failures have emerged as important challenges. Conventionally, the training and evaluation protocol of deep models follow the 'closed-world' assumption where the test data is i.i.d. with respect to the training distribution (Jiang et al. 2019). However, when such assumptions change, deep models are often brittle and can fail for a variety of reasons. For instance, models suffer a significant drop in accuracy when faced with *covariate* shifts (i.e., same input obtained from different sensors). Moreover, recent studies (Joshi, Pan, and He 2022) have revealed that deep networks are susceptible to subpopulation shifts (Y. Yang et al. 2023). These shifts occur when the training dataset is not representative of the entire population, leading to poor model performance on certain subgroups. Prominent examples include spurious correlations (Geirhos et al. 2020), where certain attributes in the training data are spuriously correlated with the labels, and class imbalance where the proportion of training images in a class is significantly larger than another. Lastly, failures can stem from limitations of the model specifications themselves and the non-generalizable decision rules learned due to the unknown biases in the training data.

Our work focuses on the critical problem of detecting samples that can lead to model failure. Key efforts in this direction utilize uncertainty estimation techniques to identify potential failures, including approaches that leverage the confidence of model

predictions (Hendrycks and Gimpel 2017; Guillory et al. 2021), energy scores (W. Liu et al. 2020), or entropy measures (Gal and Ghahramani 2016; Kirsch et al. 2021). However, these methods are fundamentally constrained by their reliance on the model's internal representations and knowledge, which ironically can be the source of failure. Given the often poorly calibrated (C. Guo et al. 2017) nature of deep models, attempting to diagnose failure solely through model predictions poses significant challenges. Although such strategies may prove effective in specific failure scenarios, they largely fall short in addressing the wide spectrum of failure modes that can manifest at test time. Moreover, they cannot typically provide interpretable insights or explanations for the observed failures, hindering our understanding and ability to rectify such issues.

In this chapter, we introduce `PRIME` (Prior Informed Model Evaluation), a holistic failure detection mechanism that is effective across a wide range of failure scenarios, without relying only on the model's predictions. By identifying the limitations of existing approaches, `PRIME` employs a strategy that leverages knowledge from foundation models such as Vision-Language Models (VLMs) (Radford et al. 2021a). To that end, we introduce a novel training protocol that aims at developing an enhanced alternative to the model by incorporating priors from foundation models, termed the Prior-Induced Model (PIM) which is the key component of `PRIME`. Unlike conventional training that directly maps images to the coarse-grained class labels, PIM learns to embed the early layer image features (Lee et al. 2022) extracted from the original model itself directly onto the VLM embedding space to harness language guidance. Here, we enforce the images to be aligned with a set of fine-grained, generic class-specific text attributes. Such a strategy guides PIM to develop decision rules that are more grounded and reliable. Our training strategy involves computing the

similarity scores between the image embeddings and these attributes and aggregate the same to estimate the coarse class-level predictions. Post-training, we examine the differences between the predictions of the given model with that of PIM and demonstrate that the prediction discrepancies can serve as an effective indicator of model failure.

Moreover, the priors introduced by the VLM in PIM allow us to dissect and understand the failure modes of the given model. By analyzing which specific attributes contribute to the predictive decisions, we provide explanations of the underlying reasons for failure. We validate our hypothesis through a comprehensive suite of benchmarks that capture different sources of model failure including spurious correlations, image corruptions, and distribution shifts. Our empirical study reveals that we significantly outperform all the considered baselines across these failure scenarios.

5.1 Related Work

Failure Detection. Failure detection in classification tasks identifies incorrect predictions by a model (Hendrycks and Gimpel 2017; F. Zhu et al. 2022; Qu et al. 2022). This problem ultimately boils down to identifying an appropriate metric or a *scoring function* that can delineate failed samples from successful ones. Early work involves using simple scores directly derived from the predictions of the model such as Maximum Softmax Probability (MSP) (Hendrycks and Gimpel 2017), predictive entropy (Kirsch et al. 2021) and energy (W. Liu et al. 2020) to identify failed samples. More recent work focuses on scores that quantify failure by evaluating the local manifold smoothness (Ng et al. 2022) around a given sample and those that are

based on agreement of a sample between different components of an ensemble (Jiang et al. 2022; Trivedi, Koutra, and Thiagarajan 2023).

However, such metrics can become unreliable to characterize failure as the model used to derive them can be potentially mis-calibrated and unreliable (C. Guo et al. 2017; Minderer et al. 2021), which necessitates the requirement to rely on external knowledge for e.g., foundation models to validate the predictions of the model. Failure detection has also been studied under the lens of generalization gap estimation (Guillory et al. 2021; Narayanaswamy et al. 2022) where the goal is to predict the accuracy of the model on an unlabeled target distribution using distributional metrics derived from a number of calibration datasets. In this work, we focus on failure evaluation in scenarios where the input samples share the same label space as that of the training data distribution.

Failure Detection with Vision Language Foundation Models. Visual-Language Models (VLMs) (Radford et al. 2021b; J. Li et al. 2022) are pre-trained on a large-corpora of image-text captions using a self-supervised objective. VLMs facilitate flexible adaptation to downstream tasks through zero-shot transfer or fine-tuning, demonstrating enhanced performance in areas such as classification and zero-shot OOD detection (Y. Wei et al. 2023; Wortsman et al. 2022; Goyal et al. 2023; Ming et al. 2022; H. Wang et al. 2023; Michels et al. 2023; Esmaeilpour et al. 2022).

Recently, such VLMs have been used as a lens to understand the failure modes and weaknesses of any pre-trained model. For instance, the authors of (Jain et al. 2023) fit a post-hoc failure detector directly on the shared VLM space to estimate whether a sample would have been correctly identified or not by the pre-trained classifier. The detector is then used to identify the directions of classifier failure modes. However, this approach requires a well-curated calibration set to fit the detector which is often

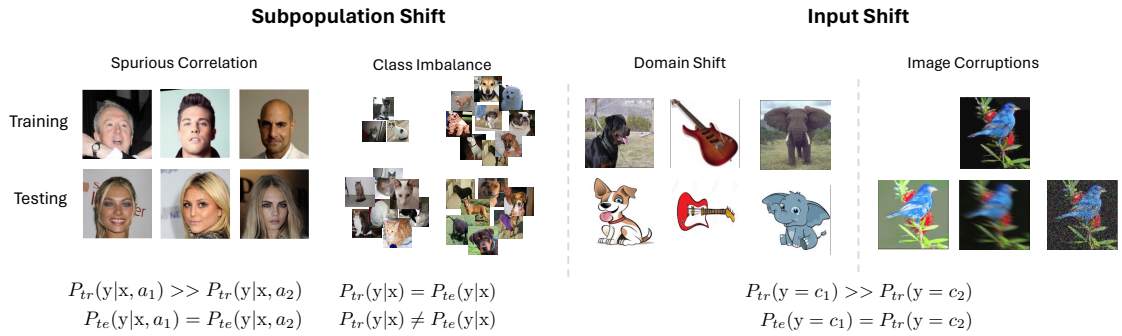


Figure 24. A visual illustration of the different failure scenarios we consider. These include scenarios when the model relies on spurious correlations present in the data i.e., when an attribute is spuriously correlated with the label (e.g., color of hair and gender). Another cause of failure is when the training data has class imbalance, leading to poorer generalization on images from the under-sampled class. Lastly, another important cause of failures are when the distribution of the test data is different from the training data. This can range from natural image corruptions to covariate shifts.

not available in practice. On the other hand, the authors of (Deng, Xiong, and Hooi 2023) demonstrate that the latent space agreement between the pre-trained model and the VLM is a potential indicator for failure. In contrast, our work aims to perform failure detection by first designing an improved classifier leveraging the VLM latent space and assessing the agreement between the classifier and its enhanced version while providing explanations for failure.

5.2 Background

Preliminaries. Let $\mathcal{F}$ denote a multi-class classifier with parameters θ, trained on a dataset $\mathcal{D} = (\mathrm{x}_i, \mathrm{y}_i)_{i=1}^{M}$ comprising M samples. Here, $\mathrm{x}_i \in \mathcal{X}$ is a 3 channel, input RGB image, and $\mathrm{y}_i \in \mathcal{Y}$ is the corresponding label, where $\mathcal{Y}$ is defined as the set $\mathcal{Y} = \{1, 2, \ldots, C\}$. Here, C denotes the total number of distinct classes. The classifier $\mathcal{F}$ operates on the input to produce the logits $\mathcal{F}(\mathrm{x})$ corresponding to every class which

is followed by a `softmax` operation to estimate output probabilities $p(\mathrm{y} = c|\mathrm{x})$ where c corresponds to the class index.

In this work, we consider the problem of failure detection in classification models, where the source of failure arises due to the following scenarios (Fig. 24) - (i) Input level shifts where the training and test images share identical conditional output distributions, i.e., $P_{tr}(\mathrm{y}|\mathrm{x}) = P_{te}(\mathrm{y}|\mathrm{x})$ but different input marginals $P_{tr}(\mathrm{x}) \neq P_{te}(\mathrm{x})$. Here, the test data can corresponds to domain variations or image corruptions. (ii) Sub-population shifts (a) Spurious correlation where the labels are non-causally associated (Y. Yang et al. 2023) with certain input characteristics or attributes in the training data over others leading to learning non-generalizable decision rules. For instance, if a_1 and a_2 correspond to two attributes of an image x, during training $P_{tr}(\mathrm{y}|\mathrm{x}, a_1) >> P_{tr}(\mathrm{y}|\mathrm{x}, a_2)$ which can fail during test time when $P_{te}(\mathrm{y}|\mathrm{x}, a_1) = P_{te}(\mathrm{y}|\mathrm{x}, a_2)$. (b) Class imbalance where the number of examples in a given class can be significantly greater than those present in another i.e., $P_{tr}(\mathrm{y} = c_1) >> P_{tr}(\mathrm{y} = c_2)$ which does not allow the classifier to optimally understand the data from class c_2, leading to sub-optimal generalization during testing.

Failure Detector Design. Failure detection is a binary classification problem of identifying whether an input sample has been correctly predicted or not by the model. We define our failure detector $\mathcal{G}$ as follows,

$$\mathcal{G}(\mathrm{x}; \theta, \tau) = \begin{cases} \text{failure}, & \text{if } \; s(\mathrm{x}; \theta) < \tau, \\ \text{success}, & \text{if } \; s(\mathrm{x}; \theta) \geq \tau. \end{cases} \tag{5.1}$$

Here, $s(.)$ is a scoring function derived from the classifier $\mathcal{F}$ that assigns higher values for correctly identified samples and vice-versa and τ is the user-controlled threshold for detection. Following standard practice in the generalization gap literature (Trivedi, Koutra, and Thiagarajan 2023; Garg et al. 2022), we identify τ such

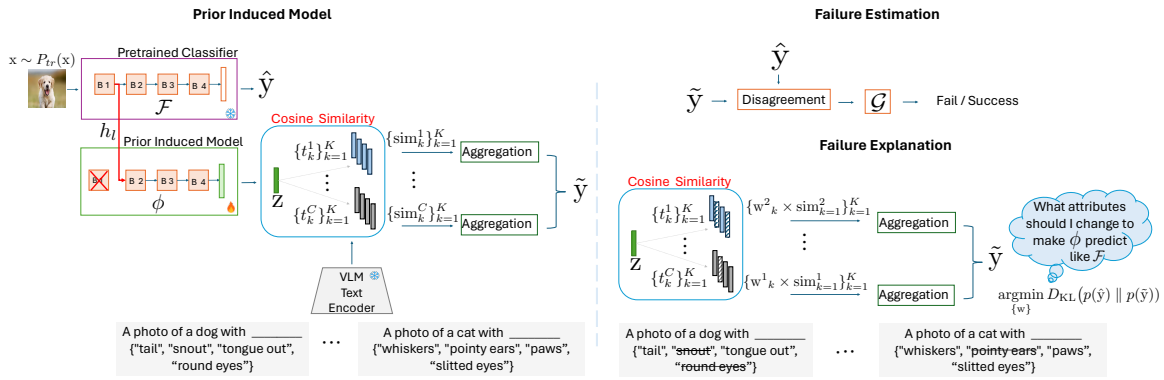


Figure 25. **Architecture of the PRIME for failure detection.** *(Left)* PRIME trains a Prior Induced Model (PIM) ϕ, identical to the architecture of the pre-trained classifier $\mathcal{F}$, utilizing priors from a VLM model. *(Top Right)* The disagreement between the predictions of ϕ and $\mathcal{F}$ serves as an indicator for failure detection. *(Bottom Right)* By adjusting attribute level weights, PRIME offers explanatory insights into the identified failures.

that $\sum_i \mathbb{I}(s(\mathrm{x}_i;\theta) \geq \tau)$ approximates the true accuracy of the held-out validation dataset.

Contrastive Language-Image Pre-training (CLIP). CLIP (Radford et al. 2021a) is a vision-language model trained on large corpus of image-text pairs with self-supervised learning. It aligns images with natural language descriptions in a shared embedding space, enabling zero-shot learning and fine-tuning for downstream tasks like image captioning (Subramanyam, Jayram, et al. 2023) and visual question answering (H. Song et al. 2022; S. Yu et al. 2024; J. Guo et al. 2023; Schwenk et al. 2022). CLIP employs image ($I(.)$) and text ($T(.)$) encoders to generate embeddings (z_I and z_T). For zero-shot inference, it computes the cosine similarity ($\cos sim$) between image and text embeddings. This similarity yields class-specific logit scores for zero-shot classification, where the prediction probability $p(\mathrm{y}|\mathrm{x})$ is calculated using `softmax`.

5.3 Proposed Approach

5.3.1 Motivation

Conventionally, a classifier $\mathcal{F}$ is trained to learn a mapping between the space of inputs $\mathcal{X}$ and the human-annotated label space $\mathcal{Y}$ by optimizing a suitable loss function (e.g., cross-entropy). By merely satisfying the training objectives, the classifier often faces the risk of learning sub-optimal decision rules between the input and the output, making it susceptible to test-time failures. This often arises due to the biases prevalent in the training data making the classifier inaccurately link the image attributes with the class labels.

Therefore, to enhance model reliability, it is crucial to train classifiers by enforcing them to focus solely on the core image characteristics associated with the label. Once such an improved classifier is trained, we can adopt a disagreement based failure detection strategy (Trivedi, Koutra, and Thiagarajan 2023) to better delineate the failed samples of the biased counterpart. A simple solution to train such classifiers is to adopt strong data augmentations (Hendrycks et al. 2022) or adversarial training (Chen et al. 2020). However, such techniques face limitations in handling the wide spectrum of failure scenarios and offering human interpretable explanations for failure.

To that end, in this section we describe our novel strategy for failure detection which involves training an enhanced classifier referred to as the Prior Induced Model (PIM) ϕ with the aid of VLM models. We believe that the prior knowledge induced by VLMs will help PIM associate relevant core attributes with the class labels. We first describe our paradigm that incorporates foundation models in classifier training. We then develop a prediction disagreement based strategy between PIM and the

biased classifier to conduct failure detection. Finally, we elucidate the capability of our approach in extracting failure explanations in order to support interpretability.

5.3.2 Incorporating Foundation Model Priors

Conventionally, classification models are trained to map the image x directly to the usually coarse label of interest y $\in \mathcal{Y}$ which many a time, encapsulates the rich core attribute characteristics describing it. For instance, in case of a dog *vs* cat classification problem, the label dog is associated with attributes such as {"wagging tail", "snout", "tongue out"} while the label cat is associated with {"whiskers", "pointy ears", "paws"} to name a few. Therefore models, due to the lack of such fine-grained information together with the biases in the training data are susceptible to rely upon trivial decision rules to make predictions. In contrast, VLMs such as CLIP offer capabilities to encode both image and textual attribute descriptions into a unified latent space that is enriched to support meaningful image-text attribute associations.

To improve the effectiveness of classification model training, we propose that aligning the model's features, when projected onto the VLM latent space, with the textual descriptions of core attributes related to the class of interest within the same latent space can enhance training. This alignment is expected to equip the model with the ability to develop decision-making rules that are both more reliable and generalizable, while also reducing the influence of existing biases.

To achieve this, we introduce the PIM model ϕ, which is guided by the VLM prior (see Fig. 25 left). The architecture of PIM closely resembles that of its counterpart $\mathcal{F}$, with the notable distinction being that its final layer projects onto the VLM latent space. This projection aims for alignment with the textual descriptions of class-level

attributes, thereby harnessing the linguistic capabilities of foundational models. PIM is specifically engineered to accept early-stage features from $\mathcal{F}$, denoted as h_l, which are then processed through PIM's analogous layers to produce the image encoding z within the VLM latent space. For instance, in the case where both $\mathcal{F}$ and ϕ are based on the ResNet architecture as described in He et al. 2016a, the output from block 1 of $\mathcal{F}$ serves as the input for block 2 in ϕ.

It must be noted that the success of our approach relies upon the quality of the fine-grained text attributes extracted for every class. While there exist strategies (Merullo et al. 2022) that are capable of extracting image-level textual descriptions, they usually involve the text decoders in the loop, which can be computationally expensive. Therefore, we resort to using Large Language Models (LLMs) to compute task-specific attribute descriptions offline.

5.3.3 Generating Task-specific Core-attribute Descriptions

LLMs (Touvron et al. 2023; Brown et al. 2020) have demonstrated their utility across a range of language tasks Radford et al. 2019; J. Wei et al. 2022; Nakano et al. 2021; Pratt et al. 2023 and are particularly adept at contextual understanding, and generating coherent text even with descriptive prompting. To extract the class-specific attribute descriptions, we query GPT-3 (Brown et al. 2020) with the prompts "List visually descriptive attributes of <CLASS>." This allows us to gather a set of K attributes $\mathcal{A}^c = \{a_k^c\}_{k=1}^K$ for every class c.

5.3.4 Training PIM

(i) Computing Cosine Similarities. We first compute the cosine similarity scores between the image embedding z produced by PIM for a given image and the text embeddings associated with attribute k from each class c. It is given by,

$$\Omega_{\mathcal{A}^c} = \{\omega_k^c\}_{k=1}^K \text{ where } \omega_k^c = \cos sim(\mathrm{z}, e_k^c) \tag{5.2}$$

Here, the text embeddings $E_{\mathcal{A}^c} = \{e_k^c\}_{k=1}^K$ for each attribute of every class are obtained using the CLIP text encoder.

(ii) Attribute Similarity Aggregation Subsequently, we aggregate these attribute similarity scores, $\Omega_{\mathcal{A}^c}$, for each class c to obtain coarse prediction logits corresponding to the class label $\mathrm{y} \in \mathcal{Y}$. We investigate two aggregation strategies namely - (i) Class-level mean and (ii) Class-level max to consolidate these scores into final class predictions which are eventually normalized using `softmax`. These strategies enable a more refined and attribute-aware determination of classification outcomes.

(iii) Optimization Objective The optimization is primarily guided by the cross-entropy loss which evaluates the discrepancy between the predicted probabilities from PIM and the ground truth label. In addition, we include consistency driven augmentations, namely CutMix (Yun et al. 2019b) and AugMix (Hendrycks et al. 2020), to improve its robustness. Additionally, we upweight the losses corresponding to the instances where (i) the biased classifier $\mathcal{F}$ predicts accurately, but ϕ does not and (ii) the biased classifier $\mathcal{F}$ does not predict accurately, as well as ϕ does not, within a training batch.

5.3.5 PRIME: Failure Estimation Using PIM

To assess the failure of the biased classifier $\mathcal{F}$, we compute the disagreement between PIM and $\mathcal{F}$ based on the discrepancy between their predictions. This disagreement score is calculated as the cross-entropy between the sample level probability distributions between the two models, with PIM being the reference distribution:

$$s(\mathrm{x}) = -\sum_{c=1}^{C} p(\mathrm{y}=c|\mathrm{x}).\log(q(\mathrm{y}=c|\mathrm{x})) \tag{5.3}$$

where $p(.)$ and $q(.)$ represent the predicted probabilities of $\mathcal{F}$ and PIM, respectively.

5.3.6 Extracting Explanations for Failure

Our failure explanation protocol is designed to elucidate the underlying reasons behind the discrepancies between predictions of $\mathcal{F}$ and ϕ. The primary objective is to identify the optimal subset of attributes necessary for aligning the PIM's prediction probabilities with those of the task model. To achieve this, we implement a strategy where we iteratively adjust a group of weights corresponding to each attribute across all classes.

Our iterative process begins by assigning an initial uniform weight to every attribute for each class within a batch. These weights are then optimized by minimizing the Kullback-Leibler (KL) divergence between the probability distributions predicted by $\mathcal{F}$ and those adjusted by PIM, accounting for the influence of the weighted attributes. Mathematically, $\underset{\{\mathrm{w}\}}{\operatorname{argmin}}\, D_{\mathrm{KL}}\big(p(\hat{\mathrm{y}}) \parallel p(\tilde{\mathrm{y}})\big)$. Here, $p(\hat{\mathrm{y}})$ represents the predictive probability distribution of $\mathcal{F}$ while $p(\tilde{\mathrm{y}})$ denotes the probability distribution of PIM. As the algorithm converges, the weights will highlight those attributes that have significant impacts on the predictions of $\mathcal{F}$, providing insights into the features

considered by it when making decisions. Figure 25 right illustrates our failure detection mechanism. We graphically illustrate the same in Fig. 28.

Algorithm 3 Training Procedure for Prior Induced Model ϕ

Require: Training data $\mathcal{D} = \{(\mathrm{x}_i, \mathrm{y}_i)\}_{i=1}^{M}$, attribute set a_k
for each class $c \in \mathcal{Y}$ extracted from LLM, VLM (CLIP) text encoder $T(.)$,
classifier $\mathcal{F}$, cross-entropy loss $\mathcal{L}(.)$, parameters ϕ initialized
with ImageNetv1 weights.
Ensure: Optimized parameters ϕ.

for each epoch **do**
  for each batch $\{(\mathrm{x}_i, \mathrm{y}_i)\}$ in $\mathcal{D}$ **do**
    Apply augmentation (Cutmix or Augmix) across batch with probability p
    Compute features $\mathbf{h}_l$ for layer l from $\mathcal{F}$.
    Use PIM to map $\mathbf{h}_l$ onto the VLM latent space to obtain $\phi(\mathbf{h}_l)$
    Initialize sample-level loss weights to uniform
    for each class c in $\mathcal{Y}$ **do**
      Compute cosine similarity between $\phi(\mathbf{h}_l)$ and the CLIP text
      embeddings $T(.)$ of attributes from class c.
      Aggregate similarities to derive class-level logits $\tilde{\mathrm{y}}$.
    end for
    Compute sample-level loss weights based on the discrepancy in predictions
    between $\mathcal{F}$ and ϕ.
    Update ϕ using the objective - $\min_{\phi} \mathcal{L}(\mathrm{y}, \tilde{\mathrm{y}})$
  end for
end for
return Optimized ϕ.

5.4 Empirical Evaluation

We conduct comprehensive evaluations of `PRIME` using various classification benchmarks and assess performance under various failure scenarios with different architectures. We employ OpenAI's CLIP ViT-B-32 model in all experiments (Radford et al. 2021b).

5.4.1 Experimental Setup

Datasets. Our experiments are centered around datasets reflecting four common sources of model failure:

- **Input-Level Shifts**: CIFAR100-C (Hendrycks and T. G. Dietterich 2019), comprising 19 types of corruptions over the CIFAR100 test images, at five severity levels, for classification into 100 categories.
- **Spurious Correlations**: (1) Waterbirds (Y. Yang et al. 2023) involves classifying images as 'water bird' or 'land bird'. The training data offers biases tied to the background (water/land). (2) CelebA (Z. Liu et al. 2015; Y. Yang et al. 2023) involves classifying if individuals have blond hair or not, with labels spuriously correlated with gender.
- **Class Imbalance**: We modify the Kaggle Cats vs Dogs dataset (Cukierski 2013), adjusting the distribution to create a training imbalance with 5,989 cat and 19,966 dog images for training, while maintaining balanced test data.
- **Distribution Shifts**: PACS (D. Li et al. 2017) includes images from four domains (Photo, Art-painting, Cartoon, Sketch), to be classified into seven categories.

Model Architectures We consider the ResNet-50 (He et al. 2016b) architecture for CelebA, DomainNet and ImageNet datasets and ResNet-18 for the remaining to train both the classifier $\mathcal{F}$ and PIM. In the Additional Results, we study the performance of `PRIME` on other architectures including ViT-B-16 (**vit**). We provide the training details in the appendix.

5.4.2 Baselines

We now define the baselines considered in work that produce sample-level scores s required for our failure detector $\mathcal{G}$. (i) Maximum Softmax Probability (`MSP`) (Hendrycks and Gimpel 2017) which is given by $s(\mathrm{x}) = \max_j p(\mathrm{y} = j|\mathrm{x})$, (ii) Predictive Entropy (`Ent`) is essentially the entropy among the predictions of a sample and is given by $s(\mathrm{x}) = -\sum_{j=1}^{K} p(\mathrm{y} = j|\mathrm{x}).\log(p(\mathrm{y} = j|\mathrm{x}))$,(iii) `Energy` (W. Liu et al. 2020) score is defined by $s(\mathrm{x}) = -T.\log\sum_{j=1}^{K}\exp^{\mathcal{F}_\theta(\mathrm{x}_j)}$. Following standard practice, we consider $T = 1$ in all our experiments. (iv) Model Agreement (`MA`) (Jiang et al. 2022; J. Chen et al. 2021) Let $\mathcal{F}_{\theta_1}, \mathcal{F}_{\theta_2} \ldots \mathcal{F}_{\theta_r}$ denote r models trained with different random seeds. Let $\mathcal{F}_{\theta_1}$ denote the base classifier. Then the score is computed as $s(\mathrm{x}) = \frac{1}{r-1}\sum_{j\neq 1}^{r}\mathbb{I}(\mathcal{F}_{\theta_1} = \mathcal{F}_{\theta_j})$,

It must be noted that we utilize negative versions of entropy and energy to reflect the fact that the samples that are correctly predicted are associated with higher scores.

5.4.3 Metrics

We consider the following metrics to evaluate failure detection performance: (i) Failure Recall (FR) which corresponds to the fraction of samples that have been correctly identified as failure, (ii) Success Recall (SR) corresponds to the fraction of samples that have been correctly predicted as successful. The trade-off between the two metrics is indicative of how aggressive or conservative the failure detector is. (iii) Matthew's Correlation Coefficient (MCC) holistically assesses the quality of the binary classification task of failure detection and provides a balanced measure when

Dataset	Method	FR	SR	MCC
CIFAR100	MSP	0.6835	0.809	0.4943
	Energy	0.6776	0.7965	0.4747
	Ent	0.6894	0.8105	0.514
	PRIME			
	+ mean	0.7949	0.7436	0.5267
	+ max	0.7933	0.7474	**0.5292**
CIFAR100-C	MSP	0.7448	0.6345	0.3593
	Energy	0.8145	0.5442	0.3577
	Ent	0.7761	0.616	0.3766
	PRIME			
	+ mean	0.8507	0.5393	0.4007
	+ max	0.8448	0.5506	**0.4015**

(a)

Dataset	Method	FR	SR	MCC
Waterbirds	MSP	0.3166	0.8891	0.2419
	Energy	0.4803	0.8047	0.2814
	Ent	0.4878	0.8022	0.2827
	PRIME			
	+ mean	0.5303	0.8310	0.3580
	+ max	0.6063	0.8580	**0.4598**
CelebA	MSP	0.4058	0.9653	0.3634
	Energy	0.4292	0.9616	0.3677
	Ent	0.4214	0.9631	0.3675
	PRIME			
	+ mean	0.5443	0.9701	**0.4928**
	+ max	0.4390	0.9621	0.3738
Cats and Dogs	MSP	0.4076	0.9235	0.3316
	Energy	0.4303	0.9196	0.3428
	Ent	0.4233	0.9212	0.3402
	PRIME			
	+ mean	0.5993	0.9468	0.544
	+ max	0.5783	0.9554	**0.5532**

(b)

Figure 26. Results on failure detection across different benchmarks - (a) CIFAR100, and image corruptions on CIFAR-100-C, and (b) subpopulation shifts from spurious correlations on Waterbirds, CelebA datasets, and class imbalance on Cats vs Dogs. PRIME consistently outperforms baselines in terms of the overall Matthew's Correlation Coefficient (MCC) as well as achieving higher failure and success recalls.

the class sizes are different. It takes into account both true and false positives and negatives respectively while assessing performance.

5.5 Training Details

5.5.1 Classifier Training

CIFAR100: Training spans 200 epochs, initial learning rate 0.1, multi-step decay at epochs 60, 120, 160, with tenfold reductions. Optimizer: SGD, momentum 0.2, weight decay 5e-4.

Waterbirds: Training spans 100 epochs, initial learning rate 0.001, multi-step decay at epochs 30, 60, with tenfold reductions. Optimizer: SGD, momentum 0.9.

CelebA: Training spans 20 epochs, learning rate 0.1. Optimizer: SGD, momentum 0.9.

Cats & Dogs: Training spans 100 epochs, initial learning rate 0.01, multi-step decay at epochs 30, 60, with tenfold reductions. Optimizer: SGD, momentum 0.9.

PACS: Training spans 200 epochs, initial learning rate 0.01, multi-step decay at epochs 30, 60, with tenfold reductions. Optimizer: SGD, momentum 0.9.

5.5.2 PIM (Prior Induced Model) Training Details

We adopt the following protocol to train PIM for all datasets. We train PIM for 200 epochs, starting with an initial learning rate of 0.1, and implement multi-step decay at epochs 60, 120, and 160, where we reduce the learning rate by factor of 0.1. We utilize the AdamW optimizer for optimization. Additionally, we apply both CutMix and AugMix transformations to the entire batch with probabilities of 0.2 each. Moreover, we carefully weight our loss function during training. The loss weights are increased by a factor of 2.0 for samples where the classifier $\mathcal{F}$ succeeds but PIM fails, and by 1.5 for cases where both the classifier and PIM fail.

5.6 Prompts Used to Query LLM (GPT3) for Attribute Generation

- **Waterbirds:** *"List 20 distinct two-word phrases that uniquely describe the visual characteristics (like type of feet, beak, wings, plumage, feathers, feather texture,*

body shape, body type etc) of {class_name}. Make sure the phrases are not long descriptions."

- **CIFAR100:** *"List 10 distinct two-word phrases that uniquely describe the visual characteristics (like shape, color, texture) of {class_name}. Make sure the phrases are not long descriptions."*
- **PACS:** *"List 10 distinct two-word phrases that uniquely describe the visual characteristics of {class_name}. Do not describe their colors. Make sure the phrases are not long descriptions."*
- **CelebA:** *"List 5 distinct two-word phrases that uniquely describe the visual characteristics of {class_name} hair person. Make sure the phrases are not long descriptions."*
- **Cats and Dogs:** *"List 10 distinct two-word phrases that uniquely describe the visual characteristics of {class_name}. Make sure the phrases are not long descriptions."*

5.6.1 Findings

Input Shifts. Fig. 26(a) showcases the results on the CIFAR100 and CIFAR100-C datasets. On the clean CIFAR100, `PRIME` outperforms the baselines with a superior MCC of 0.5292 for the max variant (versus 0.514 for the best baseline), attributed to higher failure recall (0.7933) and success recall (0.7474). On the more challenging CIFAR100-C (severity level 4), `PRIME` further highlights its efficacy by achieving an MCC of 0.4015 with max aggregation, exceeding the top baseline (negative entropy) which has an MCC of 0.3766. This is due to a balanced trade-off between failure recall (0.8448) and success recall (0.5506), distinguishing `PRIME` from other baselines that

fail to maintain such balance. These findings clearly demonstrate PRIME as robust in detecting classifier failures amid input-level shifts, surpassing other baselines in performance metrics.

Subpopulation Shifts. Our comprehensive evaluation addresses datasets affected by various subpopulation shifts. The summarized results in Fig. 26(b) underline the effectiveness of PRIME in navigating these challenges:

Waterbirds: PRIME achieves a high failure recall of 0.6063, outperforming the best baseline (negative entropy) which has a recall of 0.4878. Importantly, PRIME maintains a high success recall (0.858) with minimal compromise compared to MSP (0.8891). The outcome is a leading MCC of 0.4598, attesting to PRIME's balanced detection ability in environments with misleading background cues.

CelebA: With mean aggregation, PRIME delivers the highest MCC of 0.4928, combining a failure recall of 0.5443 with a success recall of 0.9701, showcasing its strength in addressing gender and hair color spurious correlations.

Cats vs Dogs: Exhibiting strong performance in class imbalance, PRIME (max aggregation) achieves an MCC of 0.5532, significantly surpassing the top baseline (negative energy) with an MCC of 0.3428, underlining its efficacy in balanced success and failure recall. PRIME not only demonstrates high failure detection capability but also ensures high success recall rates above 0.94, highlighting its proficiency in class-imbalanced settings.

Covariate Shifts. In this section, we evaluate the performance of PRIME in the challenging setting of identifying failures caused due to covariate shifts. We consider the PACS dataset which contains 4 different domains. We train PIM and derive individual thresholds for each of the four domains, then evaluate its performance across all domains. While we present detailed results for all baselines and metrics

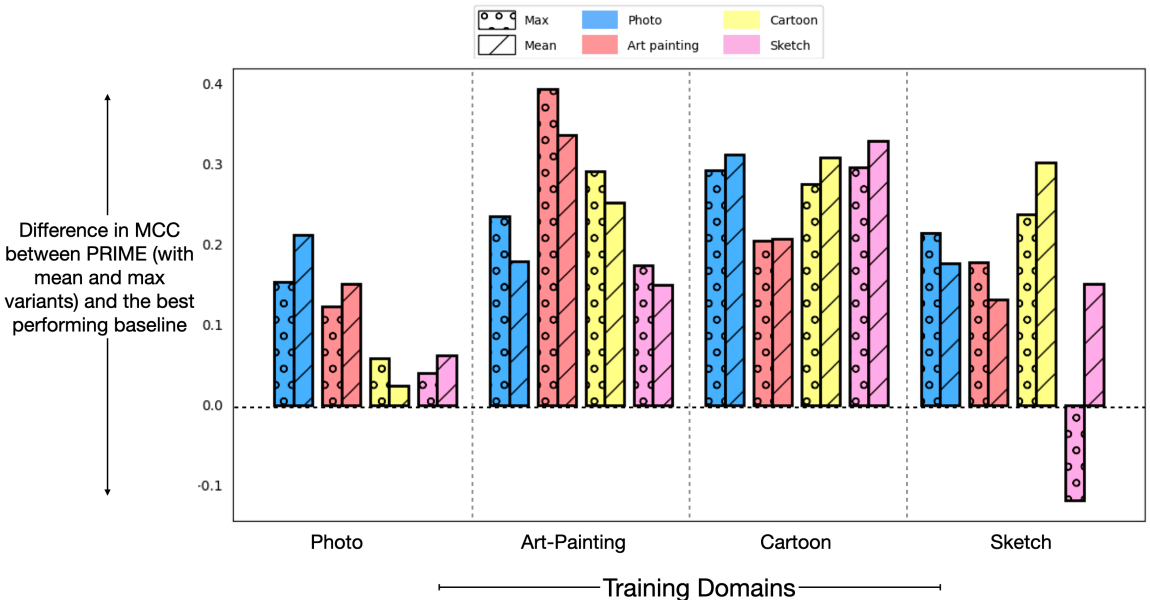


Figure 27. **PRIME produces the best performance on covariate shifts.** The bar plot provides the comparison of PRIME against the best baseline in terms of the difference in MCC on the PACS dataset involving covariate shifts across 4 different visual domains. The mean and max aggregation variants of PRIME outperform the best baseline by substantial margins across all domains.

from Fig. 14 to 17, in Fig. 27, we report the difference in MCC scores between the best performing baseline and the mean and max variants of PRIME. The X-axis lists the individual domains on which PIM has been trained and the Y-axis the difference in MCC. It can be seen from the figure that PACS outperforms the baselines by a large margin across all the domains.

In summary, across all the benchmarks when evaluated under different types of distribution shifts, including subpopulation shifts (spurious correlations, class imbalance), input shifts (image corruptions), and covariate shifts (domain variations), PRIME consistently outperforms the considered baselines by a substantial margin in terms of the overall MCC metric, failure recall and success recall. The ability of PIM to leverage language priors from vision-language models allows it to reliably detect failures stemming from spurious correlations learned by the task model, while still

maintaining high accuracy on non-failure cases. On datasets like CIFAR-100-C with severe input corruptions and on benchmarks like PACS involving significant covariate shifts across diverse visual domains, PRIME surpasses the baselines, validating its effectiveness as a generalizable failure detection approach.

5.7 Failure Explanation

Having empirically demonstrated the superior failure detection capabilities of `PRIME`, we now turn our attention to the crucial task of explaining the reasons behind failures. Since the images are projected in the VLMs multimodal embedding space, it enables us to study the impact of each attribute on the prediction outcome. By adjusting the influence of individual attributes, we ensure that the prediction probabilities generated by `PRIME` closely mirror those of the original model. This manipulation offers evidence of what attributes the task model uses.

For instance, in the top-left of Fig. 28, where the task is to correctly identify the hair color, the classifier $\mathcal{F}$ incorrectly classified the image as depicting a blond individual, while PIM accurately identified the hair color. Thus, to understand this failure, we align the probability distribution of PIM with that of $\mathcal{F}$. We do so by posing this as an optimization problem where we seek to identify the relative weights or the influence of attributes to match the probability distributions. We observe that, in this case, we achieve that goal by reducing the influence of attributes such as "Browning Tresses" and "Red Highlights". This manipulation serves as evidence that the biased classifier may not have considered these crucial attributes in its decision-making process.

Similarly, in the example shown in Fig. 29, $\mathcal{F}$ misclassifies a landbird as a waterbird

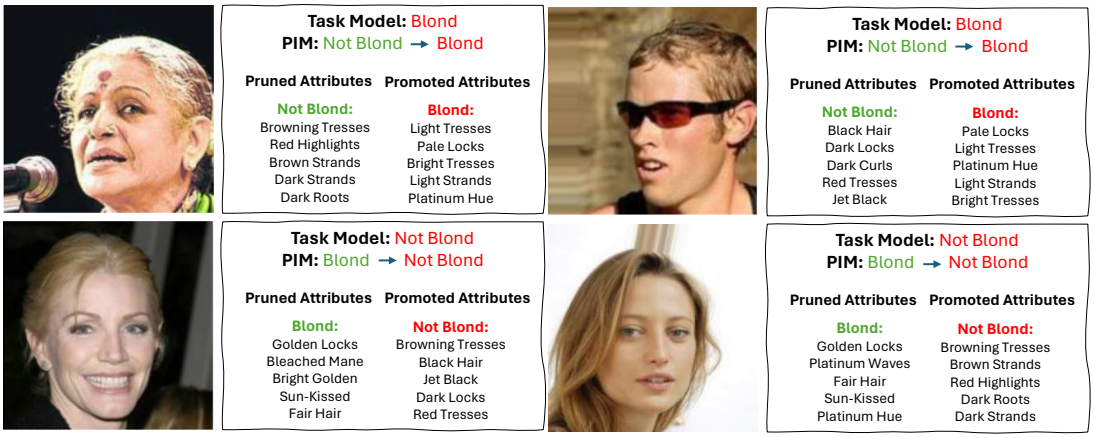


Figure 28. Example on CelebA

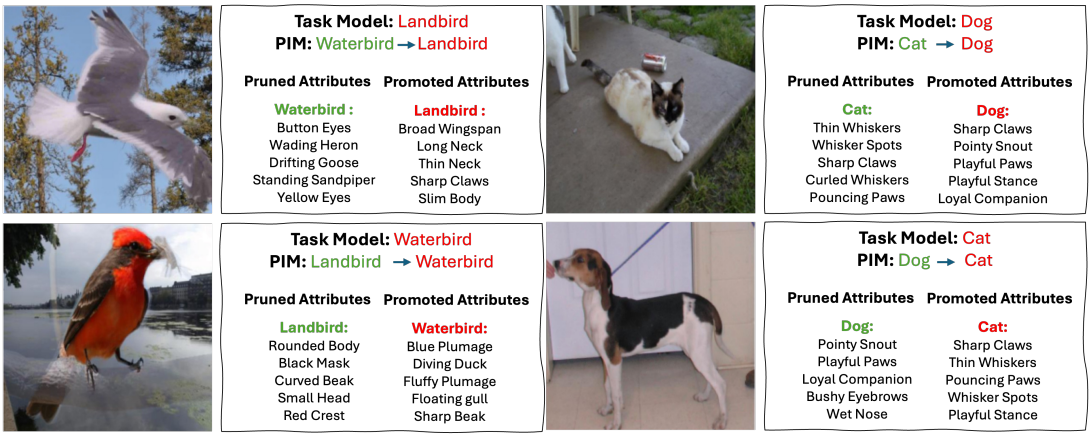


Figure 29. Example on cats vs dogs

Figure 30. **Failure Explanations**. We explain the failures of the biased classifier $\mathcal{F}$, by manipulating the influence of individual attributes in PIM, such that the prediction probabilities of PIM match that of $\mathcal{F}$. The knowledge of the attributes whose influence was needed to be reduced provides an indication that $\mathcal{F}$ has not focused on those attributes to make its decisions.

in the top left image. The most influential attribute leading to this misclassification is the broad wingspan, which could have been triggered due to the bird's flying posture commonly associated with waterbirds. These traits contributed to the erroneous classification of the landbird as a waterbird by $\mathcal{F}$.

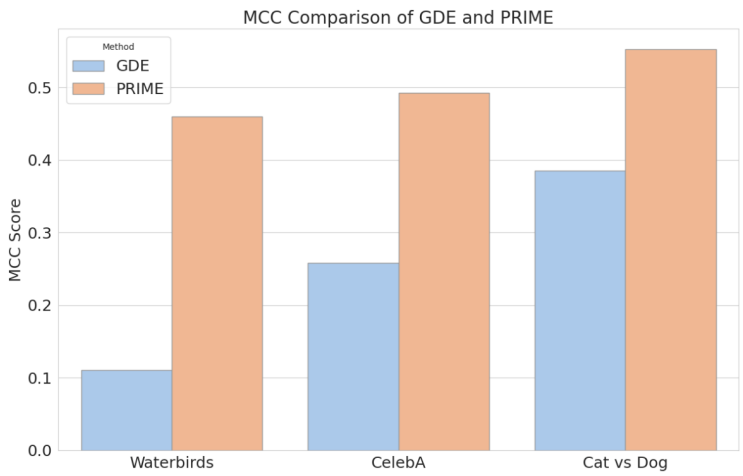
5.8 Ablations

Impact of Layer Selection of $\mathcal{F}$ on ϕ: In this study, we explore how the performance of the PIM model ϕ is influenced by the specific layer in $\mathcal{F}$ from which we extract features. This experiment uses the Resnet18 architecture, with models trained on the CIFAR100 and Waterbirds datasets. From the results presented in the table in Fig. 31, using features from the early layers (layer 1 and layer 2) of Resnet18 yields the highest MCC (Matthews Correlation Coefficient) scores. In contrast, leveraging features from the later layers leads to a noticeable decline in performance. This observation suggests that the initial layers of the network are less prone to carrying biases than the later ones, supporting the findings from previous research (Lee et al. 2022).

Model Ensembles for Disagreement Analysis: It has been shown that the prediction disagreement between different constituent members of a model ensemble can serve as an indicator of failure (Jiang et al. 2022; Trivedi, Koutra, and Thiagarajan 2023). In this experiment, we compare the failure estimation performance obtained through the disagreement between PIM and $\mathcal{F}$ to the performance obtained by the disagreement between an ensemble (GDE). To that end, we trained five different classifiers with different initial seeds on three different datasets: Waterbirds, CelebA, and Cat vs Dogs. Figure 31 evidences the superiority of the proposed approaches compared to GDE.

5.9 Additional Results

Experiment with ViT-B-16: We extend our study to incorporate the ViT architecture, specifically using the ViT-B-16 model, for the Waterbirds datasets. We



(a)

Dataset	Method	FR	SR	MCC
CIFAR100	PRIME mean			
	+ layer 1	0.7949	0.7436	0.5267
	+ layer 2	0.7670	0.7852	**0.5443**
	+ layer 3	0.6758	0.8305	0.5126
	+ layer 4	0.6761	0.8011	0.4785
	PRIME max			
	+ layer 1	0.7933	0.7474	0.5292
	+ layer 2	0.7573	0.7837	**0.5337**
	+ layer 3	0.6850	0.8297	0.5199
	+ layer 4	0.6636	0.8019	0.4680
Waterbirds	PRIME mean			
	+ layer 1	0.5303	0.8310	**0.3580**
	+ layer 2	0.3667	0.7707	0.1350
	+ layer 3	0.3640	0.7581	0.1186
	+ layer 4	0.4401	0.8208	0.2628
	PRIME max			
	+ layer 1	0.6063	0.8580	**0.4598**
	+ layer 2	0.5197	0.8153	0.3272
	+ layer 3	0.3169	0.7424	0.0577
	+ layer 4	0.3875	0.7879	0.1740

(b)

Figure 31. (a) Comparison of PRIME against the failure detection performance obtained through disagreement between predictions from an ensemble of multiple instances of $\mathcal{F}$ on Waterbirds, CelebA and Cats vs Dogs datasets respectively. (b) Ablation study analyzing the impact of using features from different layers of the base model $\mathcal{F}$ as input to the Prior Induced Model (PIM) ϕ on CIFAR-100 and Waterbirds datasets.

provide these results in Table 13. For ViT-B-16, we explore two PIM variations: one using features from the first layer and another from the ninth layer of the ViT-B-16 classifier model. From Table 13, it is evident that PRIME continues to outperform as a more reliable failure estimator, indicating its adaptability with various classifier architectures. Moreover, obtaining features from the initial layers of the classifier for constructing the PIM (Prior Induced Model) proves to be more effective than sourcing them from the deeper layers, aligning with our previous findings.

Detailed Results with PACS: Expanding on the results provided above, we provide the failure detection performance metrics under the settings where classifier $\mathcal{F}$ and PIM ϕ are trained on different domains. For all experiments, we used early layer features of the classifier. For each of these experiments, the failure estimation threshold is established based on the validation set from the respective training domain. The additional results are tabulated in Tables 14 through 17.

Table 13. Performance Comparison for ViT-B-16 architecture on the Waterbirds dataset

Dataset	Method	FR	SR	MCC
Waterbirds	MSP	0.1587	0.9048	0.0954
	Energy	0.4924	0.6732	0.1656
	Ent	0.3076	0.8301	0.1613
	PRIME layer1			
	+ mean	0.5592	0.7743	0.3416
	+ max	0.6056	0.8091	0.4235
	PRIME layer9			
	+ mean	0.4818	0.6789	0.1613
	+ max	0.5380	0.7542	0.2970

5.10 Conclusion

In this work, we introduced PRIME, a novel approach that leverages vision-language foundation models to detect failures in pre-trained image classification models. Our key insight was to train an improved version of the pre-trained classifier, PIM, that learns robust associations between visual features and class-level attributes by projecting into the shared embedding space of a VLMs such as CLIP. By analyzing the disagreement between PIM's predictions and the original biased model, PRIME can reliably identify potential failures while offering human-interpretable explanations. Extensive experiments across multiple benchmarks evidences the consistent superiority of PRIME over baselines, achieving substantially higher overall scores and better trade-offs between failure and success recalls. Our work highlights the promise of integrating vision-language priors into model failure analysis pipelines to facilitate more reliable and trustworthy deployment of vision models in safety-critical applications. Extending PRIME to other vision-language models and exploring its application to other failure modes such as adversarial attacks constitute our future work.

Table 14. Performance Comparison on PACS dataset, where the classifier and the PIM are trained and calibrated on the *Art Painting* domain

Eval. Domain	Method	FR	SR	MCC
Art Painting	MSP	0.7345	0.7799	0.4564
	Energy	0.6381	0.7698	0.3659
	Ent	0.6959	0.7837	0.4294
	PRIME			
	+ mean	0.7516	0.9822	0.7928
	+ max	0.8458	0.9784	0.8498
Cartoon	MSP	0.5636	0.6675	0.2204
	Energy	0.5033	0.7188	0.2141
	Ent	0.5799	0.6687	0.2371
	PRIME			
	+ mean	0.8394	0.6430	0.4895
	+ max	0.8945	0.6027	0.5284
Photo	MSP	0.5660	0.7857	0.3617
	Energy	0.5406	0.8265	0.3851
	Ent	0.5305	0.8254	0.3743
	PRIME			
	+ mean	0.6942	0.8594	0.5637
	+ max	0.7754	0.8424	0.6200
Sketch	MSP	0.6412	0.6088	0.2445
	Energy	0.3252	0.8238	0.1639
	Ent	0.6001	0.6252	0.2196
	PRIME			
	+ mean	0.8642	0.4977	0.3944
	+ max	0.9066	0.4576	0.4187

Table 15. Performance Comparison on PACS dataset, where the classifier and the PIM are trained and calibrated on *Cartoon* domain

Eval. Domain	Method	FR	SR	MCC
Art Painting	MSP	0.4988	0.6999	0.1938
	Energy	0.5467	0.6335	0.1739
	Ent	0.4772	0.7118	0.1855
	PRIME			
	+ mean	0.6602	0.7556	0.4011
	+ max	0.6270	0.7849	0.3977
Cartoon	MSP	0.6280	0.9206	0.4343
	Energy	0.5427	0.9211	0.3761
	Ent	0.5061	0.9349	0.3818
	PRIME			
	+ mean	0.6341	0.9950	0.7430
	+ max	0.5854	0.9950	0.7092
Photo	MSP	0.4561	0.7660	0.2281
	Energy	0.4819	0.7418	0.2266
	Ent	0.4355	0.7974	0.2431
	PRIME			
	+ mean	0.6656	0.8916	0.5552
	+ max	0.6316	0.9016	0.5354
Sketch	MSP	0.6033	0.6570	0.2497
	Energy	0.5668	0.7372	0.2926
	Ent	0.5470	0.7202	0.2575
	PRIME			
	+ mean	0.7604	0.8871	0.6220
	+ max	0.7132	0.9006	0.5887

Table 16. Performance Comparison on PACS dataset, where the classifier and the PIM are trained and calibrated on *Photo* domain

Eval. Domain	Method	FR	SR	MCC
Art Painting	MSP	0.5364	0.5983	0.1220
	Energy	0.6269	0.5254	0.1399
	Ent	0.5658	0.5847	0.1365
	PRIME			
	+ mean	0.6272	0.6955	0.2913
	+ max	0.5653	0.7266	0.2630
Cartoon	MSP	0.43532	0.56802	0.00258
	Energy	0.59117	0.51313	0.08078
	Ent	0.44831	0.55131	-0.00029
	PRIME			
	+ mean	0.47292	0.66274	0.10499
	+ max	0.42448	0.75236	0.13940
Photo	MSP	0.5278	0.9835	0.4537
	Energy	0.5000	0.9633	0.3189
	Ent	0.5556	0.9859	0.4965
	PRIME			
	+ mean	0.7143	0.9939	0.7082
	+ max	0.6571	0.9927	0.6498
Sketch	MSP	0.2226	0.8689	0.0886
	Energy	0.3440	0.9324	0.2377
	Ent	0.2176	0.8919	0.1077
	PRIME			
	+ mean	0.4229	0.9424	0.2996
	+ max	0.4103	0.9263	0.2774

Table 17. Performance Comparison on PACS dataset, where the classifier and the PIM are trained and calibrated on *Sketch* domain

Eval. Domain	Method	FR	SR	MCC
Art Painting	MSP	0.3836	0.6026	-0.0112
	Energy	0.3317	0.6462	-0.0184
	Ent	0.4331	0.5513	-0.0124
	PRIME			
	+ mean	0.9156	0.1769	0.1200
	+ max	0.9710	0.1179	0.1670
Cartoon	MSP	0.4536	0.6892	0.1326
	Energy	0.5270	0.6129	0.1279
	Ent	0.5215	0.6633	0.1692
	PRIME			
	+ mean	0.8830	0.5640	0.4717
	+ max	0.8563	0.5338	0.4065
Photo	MSP	0.3107	0.6667	-0.0179
	Energy	0.2750	0.7074	-0.0145
	Ent	0.3479	0.6481	-0.0031
	PRIME			
	+ mean	0.9679	0.1333	0.1734
	+ max	0.9850	0.1148	0.2116
Sketch	MSP	0.6822	0.9532	0.4221
	Energy	0.3458	0.9314	0.1702
	Ent	0.6449	0.9464	0.3778
	PRIME			
	+ mean	0.4673	0.9950	0.5729
	+ max	0.4299	0.9639	0.3034

Chapter 6

CONTRASTIVE KNOWLEDGE-AUGMENTED META-LEARNING FOR FEW-SHOT CLASSIFICATION

Learning to solve new tasks using only few-shot examples is a long-standing challenge. Meta-learning forms an important class of few-shot learning algorithms that leverages transferable priors from previously observed tasks to learn new tasks quickly. For example, model-agnostic meta-learning (MAML) approaches (Finn, Abbeel, and Levine 2017; Yoon et al. 2018; Lee and Choi 2018; Finn, Xu, and Levine 2018; Finn and Levine 2017) attempt to learn a single *meta* model (or base learner) on a set of observed tasks, which is assumed to be only a few gradient descent steps away from good task-specific models. Their success hinges on the assumption that the observed tasks are realizations from a common task distribution $p(\mathcal{T})$. Despite its mathematical tractability, the premise of using a single base learner can be insufficient when $p(\mathcal{T})$ is heterogeneous, *i.e.*, the degree of similarity between tasks can be vastly different (Vuorio et al. 2019). This motivates the need for a meta-model to selectively utilize knowledge from its previous experience that is the most relevant for the target task. In this context, task-aware modulation (e.g., MuMo-MAML (Vuorio et al. 2019)) is a popular principle to improve MAML on heterogeneous tasks. Conceptually, these approaches use latent task encodings, which characterize realizations from a heterogeneous task distribution, to modulate the base learner and thus improve the adaptation performance on diverse tasks.

In a quest to further improve the performance, recent methods, such as HSML (Yao et al. 2019) and ARML (Yao et al. 2020), learn an external knowledge structure for

Dataset Setting	Few-Shot Task Adaptation	Multi-Domain Few-Shot Task Adaptation	Few-Shot Dataset Generalization
TRAIN (Episodes)	$\mathcal{D}^{tr} = \{(\mathrm{x}_i, \mathrm{y}_i)\}_{i=1}^{\lvert\mathcal{D}^{tr}\rvert}$ $\mathrm{y}_i \in \mathcal{C}^{tr}$	$\mathcal{D}^{tr} = \mathcal{D}_1^{tr} \cup \mathcal{D}_2^{tr} \cdots \cup \mathcal{D}_M^{tr}$ $\mathcal{D}_m^{tr} = \{(\mathrm{x}_i, \mathrm{y}_i)\}_{i=1}^{\lvert\mathcal{D}_m^{tr}\rvert}, \mathrm{y}_i \in \mathcal{C}^{tr}$	$\mathcal{D}^{tr} = \mathcal{D}_1^{tr} \cup \mathcal{D}_2^{tr} \cdots \cup \mathcal{D}_M^{tr}$ $\mathcal{D}_m^{tr} = \{(\mathrm{x}_i, \mathrm{y}_i)\}_{i=1}^{\lvert\mathcal{D}_m^{tr}\rvert}, \mathrm{y}_i \in \mathcal{C}_m^{tr}$
TEST $\mathcal{T} = (\mathcal{S}_\mathcal{T}, \mathcal{Q}_\mathcal{T})$	$\mathcal{S}_\mathcal{T} = \{(\mathrm{x}_1, \mathrm{y}_1), \cdots, (\mathrm{x}_{kN}, \mathrm{y}_{kN})\}$ $\mathcal{Q}_\mathcal{T} = \{(\mathrm{x}_1^*, \mathrm{y}_1^*), \cdots\}$ $(\mathrm{x}, \mathrm{y}) \in \mathcal{D}^{te}$ $\mathrm{y}, \mathrm{y}^* \in \{1, \cdots, N\} \subset \mathcal{C}^{te}$	$\mathcal{S}_\mathcal{T} = \{(\mathrm{x}_1, \mathrm{y}_1), \cdots, (\mathrm{x}_{kN}, \mathrm{y}_{kN})\}$ $\mathcal{Q}_\mathcal{T} = \{(\mathrm{x}_1^*, \mathrm{y}_1^*), \cdots\}$ $(\mathrm{x}, \mathrm{y}) \in \mathcal{D}_m^{te}, m \in \{1, \cdots M\}$ $\mathrm{y}, \mathrm{y}^* \in \{1, \cdots, N\} \subset \mathcal{C}^{te}$	$\mathcal{S}_\mathcal{T} = \{(\mathrm{x}_1, \mathrm{y}_1), \cdots, (\mathrm{x}_{kN}, \mathrm{y}_{kN})\}$ $\mathcal{Q}_\mathcal{T} = \{(\mathrm{x}_1^*, \mathrm{y}_1^*), \cdots\}$ $(\mathrm{x}, \mathrm{y}) \in \mathcal{D}_{M+1}^{te}$ $\mathrm{y}, \mathrm{y}^* \in \{1, \cdots, N\} \subset \mathcal{C}_{M+1}^{te}$

Figure 32. **Few-shot classification tasks**. Here, we formally define the different problem settings considered in this study. As we move from few-shot adaptation to few-shot dataset generalization, the problem becomes increasingly challenging and requires sophisticated task-aware modulation strategies to improve the performance of MAML.

encapsulating information across training episodes and leverage the knowledge to selectively utilize prior experience during adaptation. Though these methods are known to be effective in few-shot adaptation, their generalization under large distribution shifts (Peng et al. 2019b) and semantic disparities (Triantafillou et al. 2021) can be improved.

In this chapter, we introduce Contrastive Knowledge-Augmented Meta Learning (CAML)[4], a task-aware modulation approach, with the goal of improving the generalization of meta-learners. At its core, CAML belongs to the class of MuMo-MAML-style approaches (Vuorio et al. 2019). Though CAML is similar to state-of-the-art ARML (Yao et al. 2020) in representing few-shot tasks as prototype graphs and using knowledge graphs to encode historical experience, the task encoding scheme, optimization process and the inferencing procedure are entirely different.

Summary of contributions: (i) We propose a contrastive distillation strategy to infuse prior knowledge directly into the image embedding module, which leads to

[4]CAML codebase: https://github.com/Rakshith-2905/CAML

richer task representations and eliminates the need to perform knowledge extraction during inferencing; (ii) Building upon the improved image embeddings, we adopt a computationally cheap task encoding (average pooling) in lieu of sophisticated architectures (RNN autoencoders in (Yao et al. 2019; Yao et al. 2020)); (iii) We develop an exponential moving average-based update strategy for the knowledge structure, which leads to improved generalization of the meta learner; (iv) Using standard benchmarks (Meta-Dataset, DomainNet), we perform rigorous empirical evaluation of CAML. In particular, we consider the settings of multi-domain task adaptation (we are the first to use this setting) and dataset generalization.

Findings: (i) CAML is a computationally simpler alternative to existing structure-aware meta learners – it uses simple task encoding, is not sensitive to the choice of the image embedding architecture, and does not require knowledge extraction at test-time; (ii) Under larger degrees of heterogeneity (multi-domain), we find that CAML consistently improves upon ARML (2.4% for $1-$shot and and 2.6% for $5-$shot settings); (iii) Even in the challenging dataset generalization setting, CAML provides improvements (across 8 benchmarks from the meta dataset) of 2.1% and 3.3% in $1-$shot and $5-$shot cases.

6.1 Problem Setup

In this section, we describe the problem settings considered in this study for studying the behavior of different task-aware meta learning approaches. Figure 32 provides an overview of the formulations considered. Broadly, in few-shot classification, training tasks drawn from the distribution $p^{tr}(\mathcal{T})$ are used to learn how to adapt quickly to any of the tasks, and evaluated on previously unseen test tasks from $p^{te}(\mathcal{T})$.

Common to all these formulations is that within each of the datasets, the classes seen during training are completely disjoint from those seen during testing.

A. Few-shot Task Adaptation. In this setup, let $\mathcal{D}^{tr} = \{(\mathrm{x}_i, \mathrm{y}_i)\}_{i=1}^{|\mathcal{D}^{tr}|}$ denote the training set comprising samples x_i and their labels y_i, where $\mathrm{y}_i \in \mathcal{C}^{tr}$. In other words, all samples used for training belong to one of the classes from $\mathcal{C}^{tr}$. The goal is to learn an adaptable model using $\mathcal{D}^{tr}$ to support learning new classes with only few examples. For evaluation, we construct a series of few-shot tasks and measure the model's ability to adapt to detect novel classes from $\mathcal{C}^{te}$ (i.e., $\mathcal{C}^{tr} \cap \mathcal{C}^{te} = \emptyset$). More specifically, each $k-$shot $N-$way test episode is represented as the tuple $\mathcal{T} = (\mathcal{S}_{\mathcal{T}}, \mathcal{Q}_{\mathcal{T}})$, where the *support* set contains k examples from each of the N classes (selected from $\mathcal{C}^{te}$), i.e., $\mathcal{S}_{\mathcal{T}} \coloneqq \{(\mathrm{x}_1, \mathrm{y}_1), \cdots, (\mathrm{x}_{kN}, \mathrm{y}_{kN})\}$, $\mathrm{y}_i \in \{1, \cdots, N\}$, and the *query* set $\mathcal{Q}_{\mathcal{T}} \coloneqq \{(\mathrm{x}_1^*, \mathrm{y}_1^*), \cdots\}$ contains different test examples from the same set of N classes.

B. Multi-Domain Few-shot Task Adaptation. In many practical applications, the training examples $\mathrm{x}_i \in \mathcal{D}^{tr}$ can encompass a variety of distribution shifts. Hence, we consider a new scenario where we represent the training set as a composition of datasets from M different domains, i.e., $\mathcal{D}^{tr} = \mathcal{D}_1^{tr} \cup \mathcal{D}_2^{tr} \cdots \cup \mathcal{D}_M^{tr}$, wherein all samples (regardless of the domain) belong to a common set of classes $\mathcal{C}^{tr}$. The goal here is to learn to adapt to the tasks drawn from any of the M domains. For evaluation, both the support and query sets for a test episode $\mathcal{T}$ are drawn from any domain $m \in \{1, \ldots, M\}$, i.e., $(\mathrm{x}, \mathrm{y}) \in \mathcal{D}_m^{te}$ and the N classes are picked from a disjoint set $\mathcal{C}^{te}$ similar to the previous case.

C. Few-shot Dataset Generalization. In this challenging setting, the training set is defined as a union of M different datasets $\mathcal{D}^{tr} = \mathcal{D}_1^{tr} \cup \mathcal{D}_2^{tr} \cdots \cup \mathcal{D}_M^{tr}$, and more importantly, it is assumed that each dataset contains examples from different sets of classes $\{\mathcal{C}_m^{tr}\}_{m=1}^{M}$. As a result, the goal here is to learn to adapt to completely

different semantic concepts corresponding to each of the M datasets. For evaluation, we construct test episodes using novel unseen classes from an entirely different dataset $\mathcal{D}^{te}_{M+1}$. Denoting the set of classes in the novel dataset as $\mathcal{C}^{te}_{M+1}$, we will study how effectively one can leverage the prior to generalize to unseen datasets.

6.2 Background: Task-Aware Meta Learning

While the few-shot learning literature encompasses a wide variety of approaches, meta-learning is a popular choice (Thrun and Pratt 2012; Nagabandi et al. 2018). Existing few-shot meta-learning approaches can be broadly categorized into: 1) metric-based meta-learning frameworks (Snell, Swersky, and Zemel 2017; Koch, Zemel, Salakhutdinov, et al. 2015; Vinyals et al. 2016) that learn a metric or distance function to compare different exemplars; 2) model-based approaches where meta-learning models learn to adjust the model parameters to adapt to new tasks (Munkhdalai and Yu 2017; Santoro et al. 2016); and 3) gradient-based model agnostic meta-learning models. In particular, our work builds upon model agnostic meta-learning (MAML) (Finn, Abbeel, and Levine 2017), which is formulated below.

Given a set of episodes, $\{\mathcal{T}^{tr}_1, \cdots, \mathcal{T}^{tr}_R\}$ comprised of support and query sets ($\mathcal{T}^{tr}_i = (\mathcal{S}_{\mathcal{T}^{tr}_i}, \mathcal{Q}_{\mathcal{T}^{tr}_i})$), from the training set $\mathcal{D}^{tr}$, MAML considers the meta-learner as the initialization of a task network f, *i.e.*, θ_0, and optimizes for a well-generalized initialization θ^*_0. Formally,

$$\theta^*_0 = \arg\min_{\bar{\theta}} \sum_{i=1}^{R} \mathcal{L}(f_{\theta_i}; \mathcal{Q}_{\mathcal{T}^{tr}_i}) \tag{6.1}$$

$$= \arg\min_{\bar{\theta}} \sum_{i=1}^{R} \mathcal{L}(f_{\bar{\theta}-\alpha\nabla_\theta \mathcal{L}(\theta; \mathcal{S}_{\mathcal{T}^{tr}_i})|_{\theta=\bar{\theta}}}; \mathcal{Q}_{\mathcal{T}^{tr}_i}), \tag{6.2}$$

where the task-specific initialization θ_i is obtained using a gradient step from the

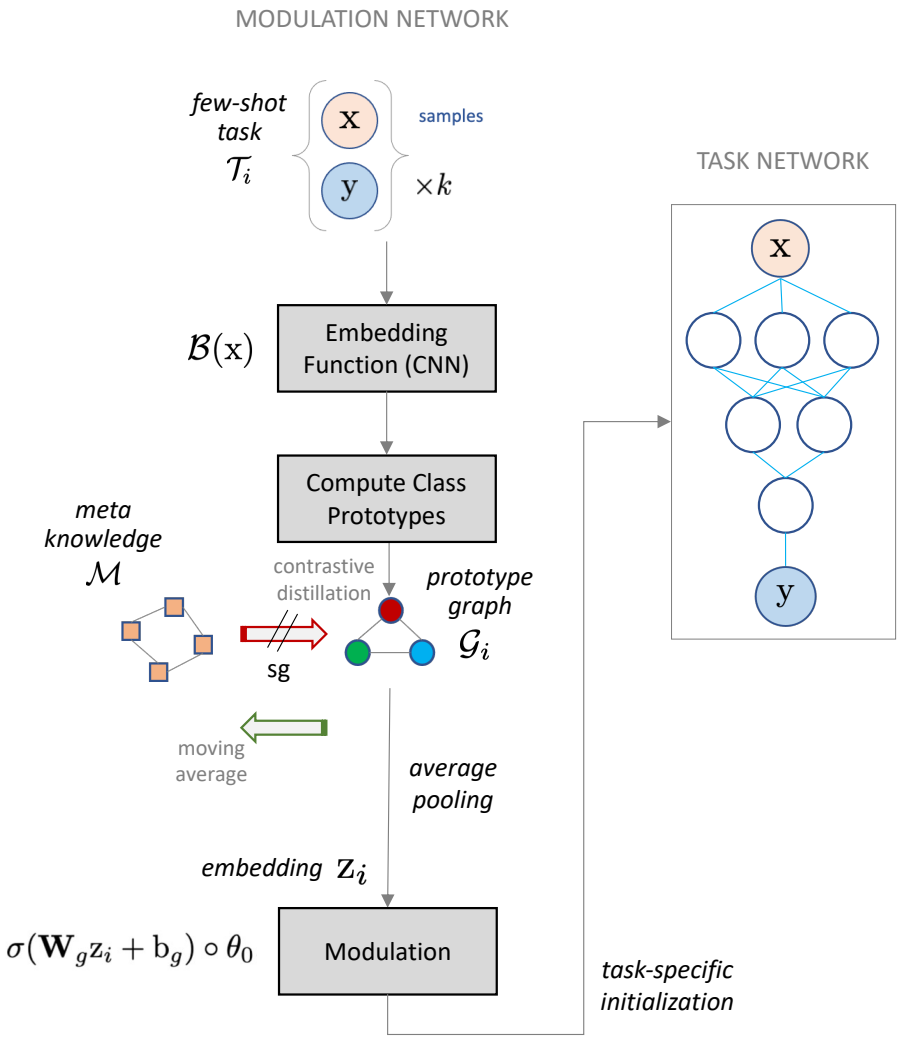


Figure 33. **Approach Overview**. An illustration of the proposed approach for task-aware meta learning. CAML involves four key steps: (i) construct a prototype graph for each training task; (ii) extract knowledge-infused task representation via contrastive distillation; (iii) modulate the base learner based on the task encoding; (iv) update the meta knowledge graph using an exponential moving average strategy. The symbol *sg* denotes the stop gradient operation, *i.e.*, the node features of $\mathcal{M}$ are not directly updated.

meta-initialization θ_0. Note, the notation $\bar{\theta}$ refers to the variables used during the optimization of this bi-level objective function. Here, $\mathcal{L}(f_\theta; \mathcal{S}_{\mathcal{T}_i^{tr}})$ is implemented as the cross entropy loss $\sum_{(\mathrm{x},\mathrm{y})\in\mathcal{S}_{\mathcal{T}_i^{tr}}} \log P(\mathrm{y}|\mathrm{x}, f_\theta)$.

Task-Aware Modulation. When the tasks used for meta-learning are sampled from a heterogeneous task distribution, inferring a common parameter initialization θ_0 for all tasks can be fundamentally restrictive. Hence, task-aware modulation (Vuorio et al. 2019) is a more effective formulation that aims at building a meta-learner which can generalize on heterogeneous task distributions through a set of latent parameters representing task-specific characteristics. For example, MuMo-MAML (Vuorio et al. 2019) first uses a task encoder to encode the training episode for a given task into a task embedding vector v_i. The task embedding is then used to obtain modulation vectors that are applied to the global initial parameters θ_0 thereby producing task-aware initialization θ_{0i}. Extending the MAML formulation in (6.2), the task-aware modulation can be carried out using the support set in the training episode $\mathcal{S}_{\mathcal{T}_i^{tr}}$ and the updated initialization θ_{0i} is used to perform the meta-optimization.

While task-specific initialization can lead to improved generalization on heterogeneous tasks, its effectiveness relies on the ability of the task embeddings to encapsulate all relationships between the large number of observed tasks. Since it is challenging to learn such expressive embeddings, more recent approaches have resorted to storage and retrieval of task-relevant information from historical experience, in order to better balance generalization and customization (task-aware modulation) (Yao et al. 2019; Yao et al. 2020). For example, hierarchically structured meta learning (HSML) and automated relational meta-learning (ARML) (Yao et al. 2020) use an external meta knowledge structure to assist the task encoding process. By adopting these knowledge-enhanced representations coupled with a sophisticated task encoder, these approaches often outperform MAML and MuMo-MAML in the standard, few-shot task adaptation setting.

6.3 Proposed Approach

Our goal is to improve the generalization of meta learners under challenging distribution shifts and large semantic disparities. To this end, we develop CAML (see Figure 1), a task-aware modulation approach that uses a meta knowledge graph $\mathcal{M}$ to encapsulate historical experience.

Overview: CAML is comprised of four key steps: (i) *Prototype graph generation*: The first step is to represent each few-shot task as a prototype graph, so that one can incorporate information from the meta knowledge graph and subsequently define a task encoding strategy. The nodes of the prototype graph correspond to class-level centroids computed using features from an image embedding module; (ii) *Knowledge-enhanced task encoding*: In this step, our goal is to enhance the node features of the prototype graph with relevant information from the knowledge graph. To this end, we propose a novel contrastive training strategy that directly refines the image embedding module by distilling from the knowledge graph. Finally, we define a task encoding based on simple average pooling of prototype node features without any learnable parameters; (iii) *Task-specific modulation*: Next, we will use the inferred task representations to compute a modulation function that can be applied to the base learner and obtain a task-specific initialization; (iv) *Meta knowledge graph update*: The final step is to update the knowledge graph in each training epoch based on the current batch of tasks, which is implemented using an exponential moving average mechanism.

6.3.1 Algorithm

Step 1: Prototype Graph Generation. Conventionally, feature extractors are used to embed data in low-dimensional latent spaces, where the different classes are easily separable. In task-aware modulation, our goal is to obtain such representations for different few-shot tasks, such that two tasks that are similar in the latent space can use the same task network initialization for effective adaptation. Each $k-$shot $N-$way training episode $\mathcal{T}_i^{tr}$ is comprised of support and query sets ($\mathcal{S}_{\mathcal{T}_i^{tr}} \mathcal{Q}_{\mathcal{T}_i^{tr}}$), wherein there are k samples in each of the N classes randomly selected from $\mathcal{C}^{tr}$. CAML begins by constructing a prototype-based graph Yao et al. 2020 with the image embeddings.

Formally, given the support set $\mathcal{S}_{\mathcal{T}_i^{tr}} \coloneqq \{(\mathrm{x}_j, \mathrm{y}_j), \forall j \in [1, \cdots, kN]\}$ for a training episode, we compute embeddings for each image x_j in the task using an embedding function. While a variety of design choices can be adopted for this, we implement the embedding function using a ResNet-18 architecture. Using the sample-level embeddings, we then compute the prototype vector for each class $n \in [1, \cdots, N]$ by taking the average of the embeddings:

$$\mathrm{v}_i^n = \frac{1}{k} \sum_{\substack{(\mathrm{x}_j, y_j) \in \mathcal{S}_{\mathcal{T}_i^{tr}} \\ y_j = n}} \mathcal{B}(\mathrm{x}_j), \tag{6.3}$$

where $\mathcal{B}$ denotes the feature extractor that projects an image x_j into $\mathbb{R}^d$. Given the sensitivity of few-shot learning methods to the limited number of examples, operating on the prototype representations reduces the effect of atypical samples. The prototype graph representation is used to both optimize the knowledge-aware task encoding and to update the meta knowledge graph. We also define a simple task encoding function based on the prototype node features:

$$\mathrm{z}_i = \Psi(\mathcal{S}_{\mathcal{T}_i^{tr}}) = \frac{1}{N} \sum_n \mathrm{v}_i^n \tag{6.4}$$

Node embeddings with high class separability and infused prior knowledge enables the use of this simple feature aggregation strategy in contrast to ARML (Yao et al. 2020) and HSML (Yao et al. 2019), which require sophisticated aggregation strategies (e.g., RNN autoencoders).

Step 2: Knowledge-Enhanced Task Encoding. The desideratum of an ideal embedding function in task-aware modulation is to produce expressive task representations that capture the complexity of a given task. Similar to existing structured meta-learning approaches, we adopt a meta knowledge graph structure to encode the historical experience and propose a novel contrastive distillation strategy to produce knowledge-enhanced task encodings. Note that existing approaches update the knowledge structure directly using gradients from the meta update step, which limits its ability to trade-off generalization and customization. Instead, we do not allow gradients to directly alter the knowledge graph (stop gradient or the symbol *sg* in Figure 1).

Formally, let us denote the knowledge graph as $\mathcal{M}$ with randomly initialized node features $\mathcal{H}_{\mathcal{M}} = \{\mathrm{h}_j\}, j = 1, \cdots, M$ and edges $\mathcal{E}_{\mathcal{M}}$. Using the prototype graph, $\mathcal{G}_i = (\mathcal{V}_i, \mathcal{E}_i)$, we perform a contrastive distillation from $\mathcal{M}$ to the embedding function $\mathcal{B}$. The edges in both $\mathcal{G}_i$ and $\mathcal{M}$ are parameterized as a function of the absolute difference of the corresponding node features. For example, for any two nodes with features a and b, $Edge(\mathrm{a}, \mathrm{b}) = \sigma(\mathrm{U}^T|\mathrm{a} - \mathrm{b}|)$, where $\mathrm{U} \in \mathbb{R}^{d \times 1}$ is the weight matrix common to all node pairs and σ is the sigmoid function.

In order to extract information for an episode $\mathcal{T}_i$ from $\mathcal{M}$, we construct a supergraph comprising nodes from both $\mathcal{G}_i$ and $\mathcal{M}$. The cross-edges are computed as the softmax of the set of negative Euclidean distances between the pairs. For a pair

$\mathrm{v}_i^n \in \mathcal{V}_i$ and $\mathrm{h}_j \in \mathcal{M}$,

$$Edge(\mathrm{v}_i^n, \mathrm{h}_j) = \frac{\exp(-\|(\mathrm{v}_i^n - \mathrm{h}_j)/\gamma\|_2^2/2)}{\sum_{\bar{n},\bar{j}} \exp\big(-\|(\mathrm{v}_i^{\bar{n}} - \mathrm{h}_{\bar{j}})/\gamma\|_2/2\big)}. \tag{6.5}$$

In order to effectively balance between knowledge-enhanced representations and the native representations from the embedding function, we propose a contrastive learning strategy inspired by several existing self-supervised learning approaches such as SimCLR and InfoNCE (Ting Chen et al. 2020; Oord, Li, and Vinyals 2018). Here, we consider the positive pair to be the task encodings from the original prototype representations and the knowledge-enhanced prototype representations obtained via neural message passing on the super-graph. The negatives are node pairs from $\mathcal{G}_i$, which indicate the level of class separability. This objective $\mathcal{L}_{CKD}(\mathcal{T}_i)$ can be expressed as:

$$-\mathbb{E}\left[\log \frac{\exp(\texttt{sim}(\mathrm{z}_i, \hat{\mathrm{z}}_i))}{\exp(\texttt{sim}(\mathrm{z}_i, \hat{\mathrm{z}}_i)) + \sum \exp(\texttt{sim}(\mathrm{v}_i^m, \mathrm{v}_i^n))}\right]. \tag{6.6}$$

Here, $\hat{\mathrm{z}}_i = \Psi[NMP(\mathcal{G}_i, \mathcal{M})]$ indicates the knowledge-enhanced task representations obtained by first performing neural message passing (NMP) on the super-graph and then subsequently averaging the updated prototype node representations $\hat{\mathrm{v}}_i^n$. Note that, when performing NMP to obtain knowledge- enhanced task representations, we do not allow the node features in the meta knowledge graph h_j to be changed, and only the prototype representations are updated. Furthermore, the similarity function `sim` is implemented using the cosine similarity. In effect, this attempts to modify the embedding function such that the task encoding is consistent with $\mathcal{M}$ while also maximizing the inter-class separability, thus producing rich task representations.

Step 3: Task-Specific Modulation. The next step is to utilize the task encodings for inferring a task-specific meta initialization. To this end, the task representation z_i is used to implement the following modulation function on the global task network

Algorithm 4 Training of CAML

Input: Distribution over training tasks $p^{tr}(\mathcal{T})$, hyper-parameters α, λ
Learnable Parameters: Embedding network $\mathcal{B}$, task network $f(\theta_0)$, modulation parameters Γ, meta knowledge graph $\mathcal{M}$, NMP network
Initialization: Randomly initialize parameters θ_0, $\mathcal{B}$, $\mathcal{M}$, and NMP network
while *not done* **do**
    Sample a batch of tasks $\mathcal{T}_i^{tr} \sim p^{tr}(\mathcal{T})$ **for** *each* $\mathcal{T}_i^{tr}$ **do**
        Sample $\mathcal{S}_{\mathcal{T}_i^{tr}}$ and $\mathcal{Q}_{\mathcal{T}_i^{tr}}$ from $\mathcal{T}_i^{tr}$
        Randomly initialize learnable edges of $\mathcal{M}$
        Compute prototype vectors $\mathcal{V}_i$ as in (6.3)
        Build prototype graph $\mathcal{G}_i$
        Construct task representation z_i from (6.4)
        Compute $\mathcal{L}_{CKD}(\mathcal{T}_i)$ using (6.6)
        Perform task-aware modulation using (6.7)
        Update $\theta_0^* = \theta_0 - \alpha \nabla_\theta \mathcal{L}(\theta; \mathcal{S}_{\mathcal{T}_i^{tr}})$
        Minimize the objective in (6.8) and update θ_0, $\mathcal{B}$, Γ, edge weights of $\mathcal{M}$, and NMP network **for** *each* $\mathcal{T}_i^{tr}$ **do**
            Obtain $\hat{\mathcal{H}}_{\mathcal{M}}^i$ using the strategy in Step 4
        Update $\mathcal{M}$ using $\hat{\mathcal{H}}_{\mathcal{M}}$ averaged over $\mathcal{T}_i^{tr}$

initialization θ_0:

$$\theta_{0i} = \Gamma(\theta_0) = \sigma(\mathbf{W}_g \mathrm{z}_i + \mathrm{b}_g) \circ \theta_0, \tag{6.7}$$

where $\mathbf{W}_g, \mathrm{b}_g$ are learnable parameters. Using a gradient-through-gradient optimization, one can then refine the task-specific initialization θ_{0i}. We incorporate our distillation objective from Step 2 into the meta-update loss function:

$$\min_{\bar{\theta}, \Omega} \sum_{i=1}^{R} \mathcal{L}(f_{\Gamma(\bar{\theta}) - \alpha \nabla_\theta \mathcal{L}(\theta; \mathcal{S}_{\mathcal{T}_i^{tr}})|_{\theta=\Gamma(\bar{\theta})}}; \mathcal{Q}_{\mathcal{T}_i^{tr}}) + \lambda \mathcal{L}_{CKD}(\mathcal{T}_i). \tag{6.8}$$

Here, Ω corresponds to the parameters of feature extractor $\mathcal{B}$, NMP network and modulation function Γ. The hyper-parameter λ controls the influence of the contrastive distillation term in the overall objective.

Step 4: Meta Knowledge Graph Update. The final step is to update $\mathcal{M}$ with

Table 18. **Few-shot task adaptation**. Performance comparison of the proposed approach against state-of-the-art meta-learning methods. In order to demonstrate that CAML performs competitively in few-shot adaptation, we used 4 different datasets from Meta-Dataset.

Method	Bird	Texture	Aircraft	Fungi	Average
Number of Shots = 1					
Meta-SGD (Z. Li et al. 2017)	55.58 ± 1.43	32.38 ± 1.32	52.99 ± 1.36	41.74 ± 1.34	45.67
MAML (Finn, Abbeel, and Levine 2017)	53.94 ± 1.45	31.66 ± 1.31	51.37 ± 1.38	42.12 ± 1.36	44.77
MT-Net (Lee and Choi 2018)	58.72 ± 1.43	32.80 ± 1.35	47.72 ± 1.46	43.11 ± 1.42	45.59
B-MAML (Yoon et al. 2018)	54.89 ± 1.48	32.53 ± 1.33	53.63 ± 1.37	42.50 ± 1.33	45.88
HSML (Yao et al. 2019)	55.99 ± 1.41	32.51 ± 1.35	51.26 ± 1.35	42.86 ± 1.42	45.66
MuMo-MAML (Vuorio et al. 2019)	56.82 ± 1.49	33.81 ± 1.36	53.14 ± 1.39	42.22 ± 1.40	46.50
ARML (Yao et al. 2020)	59.43 ± 1.46	33.30 ± 1.30	56.20 ± 1.34	45.85 ± 1.46	48.70
Proposed	59.71 ± 1.46	35.47 ± 1.38	57.55 ± 1.37	44.97 ± 1.44	49.425
Number of Shots = 5					
Meta-SGD (Z. Li et al. 2017)	67.87 ± 0.74	45.49 ± 0.68	66.84 ± 0.70	52.51 ± 0.81	58.18
MAML (Finn, Abbeel, and Levine 2017)	68.52 ± 0.79	44.56 ± 0.68	66.18 ± 0.71	51.85 ± 0.85	57.77
MT-Net (Lee and Choi 2018)	69.22 ± 0.75	46.57 ± 0.70	63.03 ± 0.69	53.49 ± 0.83	58.08
B-MAML (Yoon et al. 2018)	69.01 ± 0.74	46.06 ± 0.69	65.74 ± 0.67	52.43 ± 0.84	58.31
HSML (Yao et al. 2019)	72.07 ± 0.71	44.71 ± 0.66	64.73 ± 0.69	53.38 ± 0.79	58.65
MuMo-MAML (Vuorio et al. 2019)	70.49 ± 0.76	45.89 ± 0.69	67.31 ± 0.68	53.96 ± 0.82	59.41
ARML (Yao et al. 2020)	71.97 ± 0.70	47.18 ± 0.78	73.63 ± 0.64	55.23 ± 0.81	62.00
Proposed	73.09 ± 0.73	48.62 ± 0.69	72.88 ± 0.64	56.11 ± 0.81	62.675

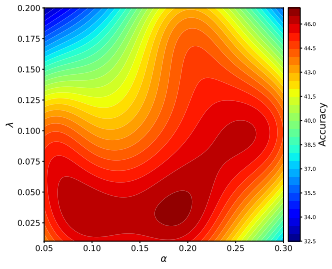
information from the current batch of tasks. By not allowing gradients from the meta update step to alter node features $\mathcal{H}_{\mathcal{M}}$, we are able to better control the historical experience encoded in $\mathcal{M}$. More specifically, using the dataset $\mathcal{T}_i^{tr}$ for each i in parallel, we update the node features $\mathrm{h}_j \in \mathcal{H}_{\mathcal{M}}$ using neural message passing on the super-graph to obtain $\hat{\mathrm{h}}_j^i$. In contrast to the distillation loss computation, during this NMP, we do not allow the prototype node features to be changed and update only the node features of $\mathcal{M}$. Note that, for both the prototype and the knowledge graphs, we use the edges inferred after the meta update in Step 3. Let $\hat{\mathrm{h}}_j$ denote the average of $\hat{\mathrm{h}}_j^i, \forall i$. Finally, we employ an exponential moving average update of the node features

Table 19. **Multi-Domain task adaptation**. Performance comparison of ARML and CAML when the meta-learners were trained using tasks from multiple domains. CAML produces consistently improved generalization in all settings.

Method	ClipArt	InfoGraph	Painting	QuickDraw	Average
	Number of Shots = 1				
ARML (Yao et al. 2020)	47.46 ± 1.48	30.61 ± 1.26	40.26 ± 1.41	65.71 ± 1.33	46.01
Proposed	50.60 ± 1.42	34.13 ± 1.35	43.13 ± 1.44	65.75 ± 1.35	48.40
Number of Shots = 5					
ARML (Yao et al. 2020)	66.58 ± 0.73	46.19 ± 0.76	56.86 ± 0.72	83.14 ± 0.55	63.19
Proposed	68.47 ± 0.71	50.35 ± 0.75	60.94 ± 0.70	83.47 ± 0.57	65.80

Table 20. **Dataset Generalization**. The evaluation is carried out using a leave-one-out protocol on the meta-dataset. We find that CAML achieves significantly improved performance over ARML.

Method	Bird	Texture	Aircraft	Fungi	Flower	Traffic	Omniglot	Quickdraw	Imagenet	Average
				Number of Shots = 1						
ARML (Yao et al. 2020)	38.34 ± 1.35	27.13 ± 1.33	27.45 ± 1.23	32.85 ± 1.38	54.79 ± 1.35	39.36 ± 1.33	70.98 ± 1.24	48.02 ± 1.36	32.67 ± 1.32	41.25
Proposed	40.56 ± 1.42	28.75 ± 1.33	28.41 ± 1.24	33.73 ± 1.37	57.89 ± 1.43	44.22 ± 1.39	71.93 ± 1.19	49.62 ± 1.31	34.95 ± 1.34	43.34
				Number of Shots = 5						
ARML (Yao et al. 2020)	55.48 ± 0.80	36.49 ± 0.64	36.39 ± 0.63	44.15 ± 0.73	71.80 ± 0.68	52.69 ± 0.66	89.61 ± 0.44	66.61 ± 0.75	44.63 ± 0.72	55.31
Proposed	58.48 ± 0.72	39.78 ± 0.65	39.45 ± 0.65	45.26 ± 0.75	73.12 ± 0.69	62.18 ± 0.69	91.06 ± 0.42	68.32 ± 0.74	50.39 ± 0.73	58.67



(a) Choice of hyper-parameters

Data Generalization: 1-Shot Training

Image Encoder	Task Encoding	Use KG?	Bird	Texture	Aircraft	Traffic	Average
Shallow CNN	Avg. Pooling	×	38.18 ± 1.34	27.91 ± 1.33	27.77 ± 1.27	44.70 ± 1.32	34.64
Resnet-18	RNN Autoenc.	×	40.88 ± 1.39	28.41 ± 1.28	28.75 ± 1.25	43.44 ± 1.35	35.47
ResNet-18	Avg. Pooling	✔	39.72 ± 1.40	29.55 ± 1.35	27.39 ± 1.24	44.26 ± 1.34	35.23
ResNet-18	Avg. Pooling	×	40.56 ± 1.42	28.75 ± 1.33	28.41 ± 1.24	44.22 ± 1.39	35.48

(b) Impact of different design choices

Figure 34. **Ablations**. We used dataset generalization experiments with 1−shot training to study the impact of different design choices on the performance of CAML: (a) Sensitivity of α, λ; (b) We explored two architectures for the image encoder (shallow CNN, ResNet18), two task encoding strategies (Average pooling, RNN autoencoder) and the effect of using the inferred knowledge graph at test time.

of the meta knowledge graph via $\mathrm{h}_j = \alpha\hat{\mathrm{h}}_j + (1-\alpha)\mathrm{h}_j$, where the hyper-parameter α controls the amount of history retained from previous episodes.

Datasets. We consider two large-scale benchmark datasets to evaluate our proposed task-aware modulation approach under the three settings in Figure 32: (i) *Meta-Dataset*: This is a widely adopted benchmark (Triantafillou et al. 2020) for few-shot image classification and is comprised of multiple image-classification datasets. From this benchmark, we utilize eight datasets for our experiments - (a) CUB-200-2011 (Bird) dataset with 200 classes; (b) describable textures dataset (Texture) with 43 classes; (c) FGVC aircraft (Aircraft) dataset with 100 classes; (d) FGVCx-fungi (Fungi) dataset with 1500 classes; (e) VGG flowers (Flower) dataset containing 102 classes; (f) German traffic signs dataset (Traffic) with 43 classes; (g) Omniglot dataset with 50 classes; (h) Quickdraw dataset with 345 classes; (i) mini-Imagenet with 100 classes. We sampled 5-way few-shot tasks from these datasets for $1-$ and $5-$shot training settings respectively. In each of these datasets, we also constructed disjoint subsets of classes $\mathcal{C}^{tr}$ and $\mathcal{C}^{te}$ for training and testing. For evaluation, we constructed k-shot N-way tasks from the unseen classes $\mathcal{C}^{te}$. Note, for all experiments, the images were resized to $84 \times 84 \times 3$; (ii) *DomainNet*: This popular benchmark (Peng et al. 2019b) for domain adaptation contains images from six different domains (clip-art, info-graph, painting, quick-draw, real, and sketch) belonging to 345 classes. To ensure availability of sufficient data for creating tasks, we ignored classes with less then 50 images and used random splits of 136 and 39 classes for training and evaluation.

Experimental details: For all our experiments we utilized a meta knowledge graph with 4 nodes with 128D features. We leverage a single layer Graph Convolutional Network (GCN) with *tanh* activation for NMP. The base learner uses a 4 layer CNN with 3×3 filters and a single liner classification layer. The 1-shot algorithms were

trained for 50K iterations and the 5-shot experiments we trained for 40K iterations, both using a meta batch size 4. We utilized the Adam optimizer for the meta update step and for the inner loop, we performed 5 gradient steps using SGD.

6.4.1 Findings

CAML performs competitively in standard few-shot adaptation. In our first experiment, we evaluated the ability of CAML to adapt to novel tasks sampled from unseen classes (within the same datasets), and compared against different gradient-based meta learning approaches on the Meta-Dataset benchmark. From the results in Table 18, we clearly notice that approaches that leverage task-aware modulation, e.g., MuMo-MAML, HSML, ARML, CAML etc., consistently outperform vanilla meta-learning approaches such as MAML and Meta-SGD. Among existing task-aware modulation strategies, ARML has been known to produce state-of-the-art results on this benchmark[5]. We find that CAML performs competitively to ARML and HSML in both 1− and 5− shot training settings, while not requiring knowledge extraction at inference time. This can be attributed to the ability of CAML to capture complex task relations and to effectively distill relevant historical information into the embedding function.

CAML can handle task heterogeneity in multi-domain adaptation. To further study the performance of CAML on heterogeneous task distributions, in this experiment, we considered DomainNet, a multi-domain benchmark. While both the

[5]We used the official implementation from the authors (https://github.com/huaxiuyao/ARML) to generate all results for ARML. Even with the prescribed settings, our metrics in Table 1 were lower than those reported in their paper. A few others have also raised this issue on Github, but the authors had not responded at the time of submission.

training and testing tasks were drawn from the same collection of domains (ClipArt, InfoGraph, Painting, QuickDraw), we ensured that the set of classes $\mathcal{C}^{tr}$ and $\mathcal{C}^{te}$ were disjoint. In this setting, the increased complexity of the task distribution makes the modulation process more sensitive, when compared to the previous experiment. For simplicity, we compare CAML with the best performing task-aware modulation baseline, *i.e.*, ARML (our experiments showed CAML was better than MuMo-MAML and HSML as well). As shown in Table 19, CAML achieves performance gaps of 2.4% and 2.6% on average, in $1-$shot and $5-$shot settings respectively.

CAML produces robust task encodings for dataset generalization. Finally, the dataset generalization experiment investigates the ability of CAML to generalize to unseen datasets. The lack of apparent semantic similarity between the classes across different datasets makes this significantly harder. However, improved performance in this problem will be of the most practical value. In this experiment, we evaluated the generalization using a leave-one-out protocol, where we train the meta learner using 8 datasets in Meta-Dataset and evaluate on the ninth dataset. From Table 20, we find that CAML achieves significant performance gains in all training settings – average gains of 2.1% and 3.3% over ARML with the same experimental setup. The observed performance improvements emphasize the efficacy of our meta knowledge construction process, and the robustness of the task representations even for unseen datasets.

6.4.2 Ablations

We now discuss the impact of different design choices. (i) ***Choice of*** α ***and*** λ: Figure 34(a) illustrates the sensitivity of different choices for α and λ. While α controls the degree to which the history is retained, λ controls the penalty for the

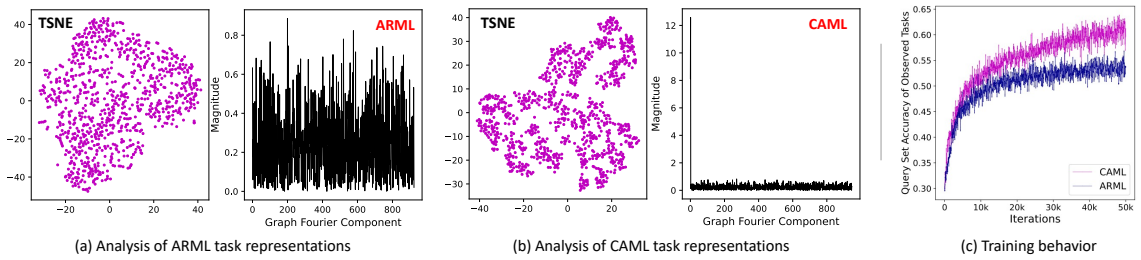


Figure 35. **Analysis.** (a)-(b) Graph Signal Analysis of the task encodings from ARML and CAML for a dataset generalization experiment. For each method, we show the 2-D TSNE embeddings of task encodings for 1000 test tasks and the graph Fourier spectrum of the accuracy score function defined at the nodes of a k-nearest neighbor graphs constructed from the task encodings (k=5); (c) Convergence characteristics of CAML and ARML for a dataset generalization experiment in the 1−shot training setting.

distillation cost. These two parameters are used to trade-off generalization (to new tasks) and customization (to observed tasks) of the learner. We find that, when α is very low, i.e., knowledge graphs evolves slowly, using a higher λ hurts the performance. On the other hand, for a reasonably higher $\alpha = 0.2$, the choice of λ becomes less sensitive. In all our experiments, we used $\alpha = 0.2, \lambda = 0.05$;

(ii) ***Choice of feature extractor***: We studied the impact of the choice of architecture for image embedding. In particular, we experimented with (a) ResNet-18; and (b) a shallow CNN (similar to (Vuorio et al. 2019)), for the case of dataset generalization. As showed in Figure 34(b), we find that the performance gap between the two models is only $\sim 0.8\%$ on average. This behavior emphasizes the flexibility of implementing CAML, wherein our contrastive distillation strategy is effective with even a shallow CNN model;

(iii) ***Choice of task encoding***: We argued earlier that, through the use of inherently effective image embeddings, CAML can work with a naïve task encoding. To validate this claim, we re-implemented CAML using RNN autoencoder-based task encodings and compared it against the average pooling strategy. Similar to the previous abla-

tion, we used a dataset generalization experiment in the 1−shot setting (see Figure 34(b)). We find that the RNN autoencoder did not lead to any significant changes in performance (on average the difference was only 0.02%);

(iv) ***Influence of using meta knowledge during adaptation***: Though we used a simple protocol for adaptation, we also experimented with a variant, where we performed knowledge infusion (using NMP) at test-time. As showed in Figure 34(b), we found that this did not provide any additional gains (on average 0.25% lower performance), thus implying that the relevant prior information has been effectively distilled into the embedding function;

(v) ***Choice of γ in Eq. 6.5***: This parameter was identified using a standard hyper-parameter search. Since the edge weights are learnable (i.e., prototype node features v are updated), we find that the choice of γ is not sensitive. We searched for γ in the range [1,12] and we noticed only marginal variations ($< 0.5\%$ on average) across choices.

6.5 Analysis

In order to justify the improved behavior of CAML over ARML, we analyzed the expressivity of their corresponding task encodings using tools from graph signal processing. More specifically, we first computed the set of task representations $\mathrm{Z}^{\mathrm{CAML}}$ and $\mathrm{Z}^{\mathrm{ARML}}$ respectively, for a set of 1000 unseen tasks from a dataset generalization experiment (*Traffic* was the unseen dataset). We also obtained the accuracies for all 1000 tasks on the query sets, which are denoted as $\mathrm{f}^{\mathrm{CAML}}$ and $\mathrm{f}^{\mathrm{ARML}}$. Our hypothesis is that if the task representations are robust, two tasks with similar encodings should lead to similar accuracy scores.

To test this hypothesis, we constructed k-nearest neighbor graphs for both CAML and ARML embeddings to obtain the graph adjacency matrices G^{CAML} and G^{ARML}. Next, we computed the graph Fourier basis. Finally, we performed the graph Fourier transform of the signal defined as a vector of accuracy scores. The expectation is that, when the task encodings are robust, the resulting graph Fourier spectrum should concentrate most of the signal's energy at low frequencies. Figure 35(a)-(b) plots the Fourier spectra obtained for CAML and ARML, when the number of neighbors k was set to 5. Even with such a small neighborhood size, the spectra for ARML contains non-trivial energy at even high frequencies, thus indicating that the task encodings are not consistent with the expected classification performance. In contrast, for CAML, we notice that most of the signal energy is concentrated at low frequencies, thereby demonstrating its improved generalization. This improved behavior is also apparent from the convergence plot in Figure 35(c). The plot shows the accuracy metric measured using the query set of each of the training tasks observed during every iteration.

Chapter 7

LEARNING KNOWLEDGE GRAPH HIERARCHIES FOR IMPROVING FEW-SHOT CLASSIFICATION

7.1 Introduction

Learning to solve new tasks using only few-shot examples is a long-standing challenge for machine-learned models. Meta-learning forms an important class of few-shot learning algorithms that leverages transferable knowledge priors from previously learned tasks to learn new tasks quickly, akin to human intelligence. For example, the widely adopted gradient-based model-agnostic meta-learning (MAML) approaches (Finn, Abbeel, and Levine 2017; Yoon et al. 2018; Lee and Choi 2018) attempt to learn a single *meta* model (or base learner) on a set of observed tasks, which is assumed to be only a few gradient descent steps away from good task-specific models. Their success hinges on the assumption that the observed set of tas are realizations from a common task distribution $P(\mathcal{T})$. Despite its mathematical tractability, the premise of using a single base learner can be insufficient when the task distribution $P(\mathcal{T})$ is heterogeneous, *i.e.*, the degree of similarity between tasks can be vastly different (Vuorio et al. 2019). For example, the observed data can contain both semantically similar (e.g. classifying sparrows and ravens) or disparate (e.g. classifying flowers and traffic signs) tasks, making the assumption of a single base learner restrictive. This motivates the need for a meta-model to selectively utilize knowledge from its previous experience that is the most relevant for the test task.

In this context, task-aware modulation (Vuorio et al. 2019) (TAM) has emerged

as an important principle to improve the performance of MAML on heterogeneous tasks. Conceptually, TAM infers latent representations to characterize realizations from a heterogeneous task distribution $P(\mathcal{T})$ (*i.e.*, learnable task representations), and modulates the base learner appropriately based on latent characteristics of the new task that we want to solve. As expected, the success of this approach hinges on the ability of the learned task representations to encapsulate all relationships between the large number of observed tasks used for MAML training (several tens of thousands), and this is known to be very challenging in practice.

In order to make fundamental advances over TAM, researchers have turned to insights from how humans effectively generalize different experiences to novel situations. One such important hypothesis is that this generalization ability relies on *relational memory* (Ngo, Newcombe, and Olson 2018; Wing et al. 2021), which allows humans to store and retrieve information based on conceptual relationships. For example, in order to learn to detect a new type of fruit, one can re-purpose knowledge about different fruits and their properties, or more generally about even shapes and colors, but not from a vehicle type classification task. This structured retrieval of information makes it easier to learn a new task by selectively utilizing knowledge from prior experience. This critical insight has led to the design of a new class of few-shot meta-learning approaches that perform structured storage and retrieval of task-relevant information. For example, hierarchical structured meta-learning (HSML) (Yao et al. 2019) uses a hierarchical knowledge structure for encapsulating task representations. Though HSML improves task-aware modulation protocol through implicit hierarchical clustering of semantics, it still uses simple vector representations for tasks similar to vanilla TAM and in practice, the design of the hierarchy (number of levels and number of clusters in each level) is highly sensitive. To mitigate these challenges, (Yao et al. 2020)

recently introduced automated relational meta-learning (ARML), which replaced task representations and the knowledge structure using more expressive graphs (instead of vectors or hierarchies) and developed a graph-based learning algorithm for automatic extraction of relational structure from heterogeneous task distributions. However, in comparison to HSML, ARML lacks the ability to organize knowledge at different levels (coarse and fine-grained), which we find to be very important for improving the expressive power of the knowledge structure.

In this chapter, we propose Structured Graph Meta-Learning SGML, which unifies the strengths of these different families of structured meta-learning methods. SGML adopts a graph-based learning framework, similar to ARML, but also organizes task-relevant priors at different levels of complexity in the form hierarchies of knowledge graphs. More specifically, each input task is represented as a graph of class prototypes and the node features in this graph are augmented with relevant historical information from the knowledge structure through a neural message passing mechanism. Finally, the prototype graph and the augmented prototype graph are aggregated to perform task-aware modulation of the shared base learner. Using the Meta-Dataset benchmark (Triantafillou et al. 2020), we carried out empirical studies to compare the proposed SGML with state-of-the-art structured meta-learning approaches, namely HSML and ARML, and we find that SGML provides significant performance gains even with highly heterogeneous tasks.

Our contributions are as follows:

- We propose SGML, a new structured meta-learning approach, which learns a hierarchy of knowledge graphs to represent information from historical tasks and performs task-aware modulation for effective adaptation.
- We empirically showcase the utility of our more expressive knowledge structure

by obtaining average performance gains of $\sim 2\%$ over state-of-the-art structured meta-learning approaches.

- We study the new scenario of out-of-distribution few-shot learning, where the model is evaluated on tasks from unseen datasets, and show that SGML's superior inductive bias improves performance by $\sim 3\%$ on novel unseen tasks when compared with state-of-the-art framework ARML.

7.2 Few-Shot Meta Learning

The goal of few-shot meta-learning is to learn task-specific functions using few data samples and training iterations. For a task $\mathcal{T}_i$ sampled from an underlying task distribution $P(\mathcal{T})$, we have $D_i^{tr} = (\mathrm{x}_j, y_j) : j \in [1, N^{tr}]$ constituting the training set, and correspondingly D_i^{ts} representing the test set. The goal is to perform a K-way classification task, where K is the number of unique labels in task $\mathcal{T}_i$. Note, while the number of classes K is typically fixed across tasks, the specific classes vary across tasks. In few-shot learning settings, the number of data examples (or *shots*) in each class is assumed to be very small. During the training phase, the model can access several tasks $\{\mathcal{T}_i\}$, from which it can "learn to learn" effectively.

While the few-shot learning literature encompasses a wide variety of training algorithms, meta-learning is a popular choice (Thrun and Pratt 2012) for both classification and reinforcement learning (Nagabandi et al. 2018). Existing few-shot meta-learning approaches can be broadly categorized into: 1) metric-based meta-learning frameworks (Snell, Swersky, and Zemel 2017; Koch, Zemel, Salakhutdinov, et al. 2015; Vinyals et al. 2016) that learn a metric or distance function to compare different exemplars; 2) model-based approaches where meta-learning models learn to

adjust the model parameters to adapt to new tasks. such as the RNN-based meta-learning of (Munkhdalai and Yu 2017) and (Santoro et al. 2016); and 3) gradient-based model agnostic meta-learning models. Our work builds upon these, specifically model agnostic meta-learning (MAML) (Finn, Abbeel, and Levine 2017), which can be formulated as follows: Given a set of R few-shot learning tasks, $\{\mathcal{T}_1, \cdots, \mathcal{T}_R\}$, sampled from $P(\mathcal{T})$, MAML considers the meta-learner as the initialization of the model f, *i.e.*, θ_0, and optimizes for a well-generalized initialization θ_0^*. Formally,

$$\theta_0^* = \arg\min_{\bar{\theta}} \sum_{i=1}^{R} \mathcal{L}(f_{\theta_i}; \mathcal{D}_i^{ts}) \tag{7.1}$$

$$= \arg\min_{\bar{\theta}} \sum_{i=1}^{R} \mathcal{L}(f_{\bar{\theta} - \alpha \nabla_\theta \mathcal{L}(\theta; \mathcal{D}_i^{tr})}; \mathcal{D}_i^{ts}), \tag{7.2}$$

where the task-specific initialization θ_i is obtained using a gradient step from the meta-initialization θ_0. Here, the loss function $\mathcal{L}(\theta; \mathcal{D}_i^{tr})$ denotes the cross entropy.

Task-Aware Modulation. When the tasks used for meta-learning are sampled from a heterogeneous task distribution, inferring a common parameter initialization θ_0 for all tasks can be fundamentally restrictive. Hence, (Vuorio et al. 2019) proposed a more effective formulation that aims at building a meta-learner which can generalize on heterogeneous task distributions through a set of latent parameters representing task-specific knowledge. First, a task encoder is used to encode the training data for a given task $\mathcal{T}_i$ into a task embedding vector v_i. The task embedding is then used to obtain modulation vectors that are applied to the global initial parameters θ_0, thereby producing task-aware initialization θ_{0i}. Extending the MAML formulation in (7.2), the task-aware modulation is carried out using the training data $\{\mathcal{D}_i^{tr}\}$ and θ_{0i} is used to perform meta-optimization.

While task-specific initialization can lead to improved generalization on heterogeneous tasks, its effectiveness relies on the ability of the task embeddings to encapsulate all relationships between the large number of observed tasks and thus performing similar parameter modulation for related tasks. Since it is challenging to learn such expressive embeddings, more recent approaches have resorted to storage and retrieval of task-relevant information from external knowledge structures in order to better balance generalization and customization (task-aware modulation). In particular, hierarchically structured meta-learning (HSML) and automated relational meta-learning (ARML) are the two such approaches that are the most relevant to the proposed work.

Hierarchically Structured Meta-Learning (HSML) (Yao et al. 2019). The core idea of HSML is to enhance few-shot meta-learning by explicitly inducing a task clustering step, thus enabling customization to different clusters of tasks as well as ensuring generalization among semantically related tasks. In particular, by adopting a hierarchical clustering formulation, HSML effectively performs coarse- and fine-grained categorization of heterogeneous tasks. Given a task $\mathcal{T}_i$, HSML begins by inferring a task embedding vector (a.k.a task representation), similar to TAM, using a recurrent autoencoder network. Subsequently, this task representation is augmented with information from the hierarchical knowledge structure using a soft cluster assignment strategy.

Each node (or cluster) in the hierarchical knowledge structure contains learnable parameters (weights and biases) that are used to transform the input task representation from level $\ell - 1$. The updated task representations obtained after traversing the entire hierarchy can then be leveraged to perform task-aware modulation.

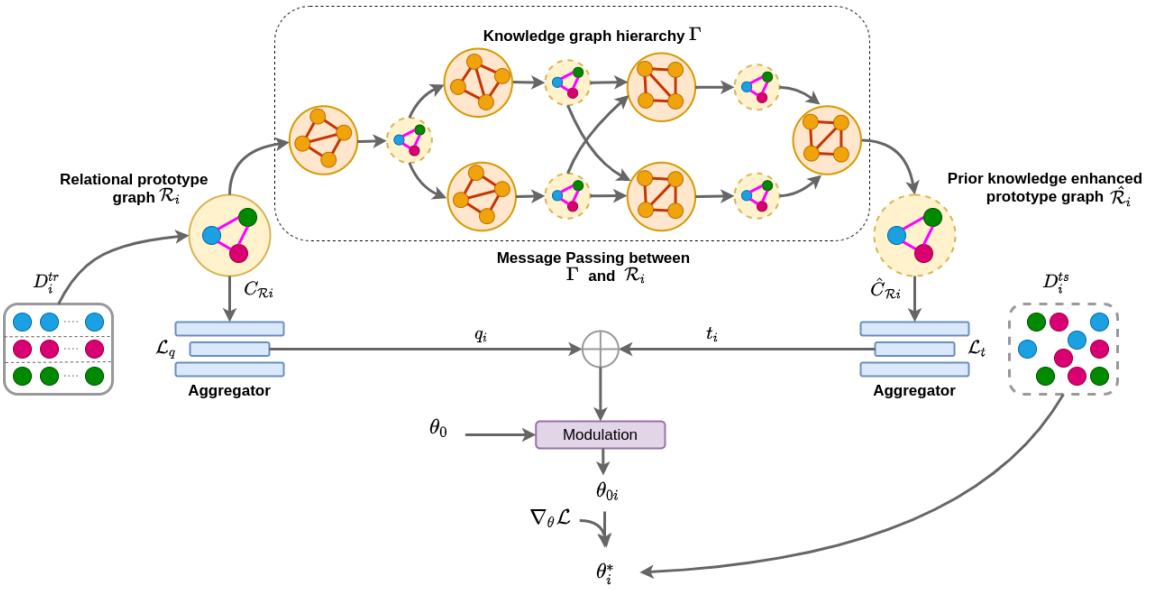


Figure 36. An overview of the proposed approach for structure-aware meta-learning. SGML represents input few-shot tasks using prototype graphs and constructs knowledge graph hierarchies to capture task relationships as the algorithms processes a sequence of tasks. For any task $\mathcal{T}_i$, its prototype graph is augmented with relevant information from the learned knowledge structure (*i.e.*, prior experience). Finally, the updated task representations are used to modulate the global meta-initialization parameters θ_0 to obtain task-specific initialization θ_{0_i}.

Automated Relational Meta-Learning (ARML) (Yao et al. 2020). While hierarchical clustering enables consistent modulation of the meta-parameters for related tasks, in practice, the design of the hierarchical structure is highly sensitive and more importantly, it relies on simple vector representations for tasks. Instead, ARML proposed to replace both task representations as well as the knowledge structure using a learnable meta-knowledge graph $\mathcal{G}$ to succinctly encode the prior knowledge as the meta-learning algorithm continues to process a sequence of tasks.

7.4 Proposed Approach

In this section, we present SGML, a gradient-based meta-learning approach for few-shot classification, which constructs a hierarchy of knowledge graphs to effectively represent task relationships. For a novel test task, SGML can automatically extract relevant information from the knowledge structure and appropriately tailor the meta-initialization parameters. While our approach belongs to the family of structure-aware meta-learning approaches such as ARML and HSML, it fundamentally differs by the structure utilized for modeling relationships between tasks.

As illustrated in Figure 36, for a given task $\mathcal{T}_i$, we construct a prototype-based graph representation (Yao et al. 2020), and distill the task-relevant knowledge from every level of the knowledge graph hierarchy in order to update the prototype graph with historical task information. We then use an aggregate of the original task representation and the enhanced prototype graph to tailor the global initialization parameters θ_0 to effectively solve the current task. In the rest of this section, we describe the proposed approach in detail.

7.4.1 Representing Tasks using Prototype Graphs

For an input task $\mathcal{T}_i$ with training samples $(\mathrm{x}_j, \mathrm{y}_j) \in D_i^{tr} | \forall j \in [1, N^{tr}]$, we first construct a prototype-based relational graph $\mathcal{R}_i$. The nodes $\mathcal{C}_i$ denote the prototypes corresponding to different classes in $\mathcal{D}_i^{tr}$ and the edges $\mathcal{E}_i$ indicate the similarities between prototypes. Note that using the class prototypes to construct $\mathcal{R}_i$ leads to more reliable task representations. As opposed to the vector embeddings used in TAM (Vuorio et al. 2019) or HSML (Yao et al. 2019), such representations are

more expressive both to capture cross-task relations as well as to retrieve relevant information from the learned knowledge structure. Following standard practice in image classification models, the prototypes are computed using features from an embedding network $\mathcal{B}$:

$$\mathrm{c}_i^k = \frac{1}{N_k^{tr}} \sum_{j=1}^{N_k^{tr}} \mathcal{B}(\mathrm{x}_j), \tag{7.3}$$

where N_k^{tr} denotes the number of samples in class k and the embedding function $\mathcal{B}$ projects an image x_j into $\mathbb{R}^d$, such that samples from the same class are closer to each other. Thus, our prototype for each class, which becomes the features for that class's node in $\mathcal{C}_i$, is the mean embedding of the images in that class. We compute edges $\mathcal{E}_i$ of the prototype graph based on the similarity between the node features:

$$\mathcal{E}_i(\mathrm{c}_i^j, \mathrm{c}_i^m) = \sigma(\|\mathrm{c}_i^j - \mathrm{c}_i^m\|_2^2), \tag{7.4}$$

where $\mathrm{c}_i^j, \mathrm{c}_i^m$ are two different nodes in $\mathcal{C}_i$ and σ denotes the sigmoid function. In our empirical studies, not including any additional learnable parameters in (7.4) leads to a more stable convergence of our meta-learning algorithm.

7.4.2 Constructing Knowledge Graph Hierarchies

To efficiently extract prior knowledge relevant to the current task $\mathcal{T}_i$, we propose to utilize knowledge graph hierarchies. While each level in the hierarchy represents historical task priors at different complexity, each knowledge graph in a level represents different task groups at a certain complexity. Formally, we represent our proposed knowledge graph hierarchies as

$$\Gamma = \{\mathcal{G}_\ell^m | \forall \ell \in [1, \ldots, L], m \in [1, \ldots, M_\ell]\} \tag{7.5}$$

where L denotes the total number of levels and M_ℓ represents the number of knowledge graphs in level ℓ.

In order to automatically infer the knowledge structure, every meta-graph $\mathcal{G}_\ell^m = (\mathcal{H}_\ell^m, \mathcal{A}_\ell^m)$ is initialized with learnable node features $\mathcal{H}_\ell^m \in \mathbb{R}^{V \times d}$, and the corresponding adjacency matrix $\mathcal{A}_\ell^m \in \mathbb{R}^{V \times V}$.

Here, V denotes the total number of vertices for each meta-graph. Similar to the prototype graph construction, the edge weights between two vertices h_p and h_q in a meta-graph $\mathcal{G}_\ell^m$ can be obtained as:

$$\mathcal{A}_\ell^m(\mathrm{h}_p, \mathrm{h}_q) = \sigma\left(\mathbf{W}_{pq}^{\ell,m}(|\mathrm{h}_p - \mathrm{h}_q|) + \mathrm{b}_{pq}^{\ell,m}\right), \tag{7.6}$$

where $\mathbf{W}_{pq}^{\ell,m}$ and $\mathrm{b}_{pq}^{\ell,m}$ represent learnable weights and biases. Note that the parameters, the node features, and edge functions are updated at meta-test time using D_i^{ts}, the (limited) labeled data used to adapt to the new task.

7.4.3 Augmenting Task Representations via Message Passing

For every input task $\mathcal{T}_i$, our goal is to update the prototype graph representation by distilling task-specific knowledge from each level of the knowledge graph hierarchy. At level ℓ, we systematically propagate information from each of the meta-graphs $\mathcal{G}_\ell^m$ to the prototype graph by constructing a super-graph $\mathcal{S}_i^{\ell,m}$ and then performing neural message passing to update node features $\mathcal{C}_i$. The vertices of the super-graph correspond to the union of vertices from $\mathcal{R}_i$ and $\mathcal{G}_\ell^m$, and existing edges within each of the two graphs are retained. We also add edges between the vertices $\mathcal{C}_i$ of the prototype graph and the vertices $\mathcal{H}_\ell^m$ of the m^{th} knowledge graph in level ℓ using a

Gaussian kernel-based link function:

$$\mathcal{Q}_i^{\ell,m}(\mathrm{c}_i^k, \mathrm{h}_p) = \frac{\exp\big(-\|(\mathrm{c}_i^k - \mathrm{h}_p)/\gamma\|_2^2/2\big)}{\sum_{v=1}^{V} \exp\big(-\|(\mathrm{c}_i^k - \mathrm{h}_v)/\gamma\|_2^2/2\big)} \tag{7.7}$$

where $\mathrm{h}_v, \mathrm{h}_p \in \mathcal{H}_\ell^m$ and γ is a scaling factor. This process results in the super-graph, denoted as $\mathcal{S}_i^{\ell,m} = (\mathcal{C}_i \bigcup \mathcal{H}_\ell^m, \mathcal{Q}_i^{\ell,m})$, comprising $K + V$ vertices in total. Following the messaging passing strategy adopted by ARML (Yao et al. 2020), we leverage a Graph Convolutional Network (GCN) (Kipf and Welling 2017) to distill information from the knowledge graph to update the task representation. After carrying out a GCN forward pass on the super-graph, the top K components of the updated feature set correspond to the prototype graph updated with prior task knowledge from $\mathcal{G}_\ell^m$. Using this approach, one can distill from each of the knowledge-graphs in level ℓ to obtain M_ℓ updated prototype graphs. Formally, for every knowledge graph ~~KG~~ in level 1, we compute

$$\hat{\mathcal{C}}_i^{1,m} = \texttt{GCN}(\mathcal{V}_i, \mathcal{G}_1^m) \tag{7.8}$$

The resulting representations from any level ℓ are subsequently used to select relevant information from the next $(\ell+1)^{\text{th}}$ level in the hierarchy. Though one can employ an attention mechanism to compute relevance of a knowledge graph n in level $\ell+1$ using the updated prototype representation at knowledge graph m in level ℓ, similar to the soft cluster assignment in HSML (Yao et al. 2019), we find that a uniform attention itself is highly effective. In other words,

$$\hat{\mathcal{C}}_i^{\ell+1,n} = \frac{1}{M_\ell} \sum_{m=1}^{M_\ell} \texttt{GCN}(\hat{\mathcal{C}}_i^{\ell,m}, \mathcal{G}_{\ell+1}^n) \tag{7.9}$$

Through this hierarchical knowledge distillation process, our approach systematically incorporates task-relevant information at different levels of complexity. After processing information in all L levels of the knowledge graph hierarchy, the final task

representation for the task $\mathcal{T}_i$ is obtained by performing average pooling of all M_L prototype representations in the last level:

$$\hat{C}_i = \frac{1}{M_L} \sum_{m=1}^{M_L} \hat{C}_i^{L,m}. \tag{7.10}$$

From our empirical studies, we make the striking observation that knowledge graph hierarchies provide a powerful way to improve the expressive power of the knowledge structures in capturing cross-task relationships. Interestingly, the performance of existing approaches such as ARML declines steadily as one arbitrarily increases the number of vertices in the meta-knowledge graph. With SGML, we can either increase the depth L of the hierarchy or alternately increase the number of graphs M_ℓ in any level to enhance its expressive power.

7.4.4 Task-Specific Modulation of Meta Parameters

The final important step of our meta-learning algorithm is to utilize the enhanced task representations for inferring task-specific meta initializations. Our implementation of this step exactly follows HSML (Yao et al. 2019) and ARML (Yao et al. 2020), where the task-relevant information from the hierarchy along with the original prototype graph are used to perform task-aware modulation. The inductive biases from the historical knowledge can enable rapid adaptation of the global meta-model to any task from a heterogeneous task distribution. To this end, we first aggregate C_i and $\hat{C}_i$ into vector representations using two RNN-based autoencoders. The RNN aggregators create dense representations for each $\mathrm{c}_i^j \in C_i$ and $\hat{\mathrm{c}}_i^j \in \hat{C}_i$. Finally, concise task embedding vectors q_i and t_i for C_i and $\hat{C}_i$, respectively, are obtained by performing

average pooling over all K vertices:

$$\mathrm{q}_i = \frac{1}{K}\sum_{j=1}^{K}(\Psi_q^{\mathrm{enc}}(\mathrm{c}_i^j)); \quad \mathrm{t}_i = \frac{1}{K}\sum_{j=1}^{K}(\Psi_t^{\mathrm{enc}}(\hat{\mathrm{c}}_i^j)) \tag{7.11}$$

In order to train these two RNN aggregators, we also include two additional reconstruction loss terms based on both the encoder-decoder networks:

$$\mathcal{L}_q = \sum_j \|\mathrm{c}_i^j - \Psi_q^{\mathrm{dec}}(\Psi_q^{\mathrm{enc}}(\mathrm{c}_i^j))\|_2, \tag{7.12}$$

$$\mathcal{L}_t = \sum_j \|\hat{\mathrm{c}}_i^j - \Psi_t^{\mathrm{dec}}(\Psi_t^{\mathrm{enc}}(\hat{\mathrm{c}}_i^j))\|_2. \tag{7.13}$$

Along with the parameters of knowlege graph hierarchy and the base learner, the two autoencoders are updated using the meta-test set D_i^{ts}.

The task representations q_i and t_i are used to implement the modulation function for obtaining task-specific initialization θ_{0i}:

$$\theta_{0i} = \sigma(\mathbf{W}_g(\mathrm{q}_i \oplus \mathrm{t}_i) + \mathrm{b}_g) \circ \theta_0, \tag{7.14}$$

where $\oplus$ denotes the concatenation operation and $\mathbf{W}_g, \mathrm{b}_g$ are learnable parameters. Using a gradient-through-gradient optimization, one can then refine the task-specific initialization θ_{0i}. In summary, we use the θ_{0i}^* (computed in the meta-train phase) to infer parameters of the knowledge graph hierarchy Γ, task embedding function $\mathcal{B}$, and the task aggregators using the following optimization objective:

$$\sum_{\mathcal{T}_i \in P(\mathcal{T})} \mathcal{L}(f_{\theta_i^*}, D_i^{ts}) + \mu_1\mathcal{L}_q + \mu_2\mathcal{L}_t, \tag{7.15}$$

where μ_1, μ_2 are user-specified hyper-parameters.

Algorithm 5 Training of SGML

Input: $P(\tau)$: Distribution over heterogeneous tasks. ℓ: number of levels in hierarchical structure Γ; M_ℓ: number of knowledge graphs in level ℓ.
Output: Learned structure Γ, embedding network $\mathcal{B}$, RNN aggregators, base learner $f(\theta_0)$
Initialization: Randomly initialize parameters θ_0, parameters of Γ, and hyper-parameters $\alpha, \gamma, \mu_1, \mu_2$
while *not done* **do**
    Sample a batch of tasks $\tau_i \sim P(\tau)$ **for** *each* τ_i **do**
        Sample D_i^{tr} and D_i^{ts} from τ_i
        Construct prototype graph $\mathcal{R}_i$ using (7.3) and (7.4)
        Extract knowledge from Γ to get updated prototype representation following (7.7)-(7.10)
        Aggregate the original and the knowledge-enhanced task representation using (7.11)
        Perform task-aware modulation using (7.14)
        Update $\theta_0^* = \theta_0 - \alpha \nabla_{\theta_0} \mathcal{L}(\theta_0; D_i^{tr})$
    Update θ_0, Γ, $\mathcal{E}$, and RNN aggregators to minimize the objective in (7.15)

7.5 Results and Findings

7.5.1 Dataset Description

We consider six benchmark image datasets to construct the few-shot classification experiments: CUB-200-2011 (Bird) with 200 classes, Describable Textures Dataset (Texture) with 43 classes, FGVC of Aircraft (Aircraft) with 100 classes, FGVCx-Fungi (Fungi)) with 1500 classes, VGG Flower (Flower) with 102 classes, and the German traffic signs dataset (Traffic Signs) with 43 classes. All datasets were obtained as part of the Meta-Dataset (Triantafillou et al. 2020), a large-scale benchmark for evaluating meta-learning algorithms. While the first four datasets are used to train and evaluate different structure-aware meta-learning approaches, the latter two are used to test the

Table 21. Performance comparison of the proposed approach against state-of-the-art meta-learning methods for few-shot classification. We show results for 4 different datasets from the Meta-Dataset benchmark. All results were obtained using 1000 test tasks in each case. Our method, SGML, achieves the most favorable average accuracy and rank across datasets.

Method	Bird	Texture	Aircraft	Fungi	Average	Average Rank
Number of shots = 1						
Meta-SGD	55.58 ± 1.43	32.38 ± 1.32	52.99 ± 1.36	41.74 ± 1.34	45.67	7.5
MAML	53.94 ± 1.45	31.66 ± 1.31	51.37 ± 1.38	42.12 ± 1.36	44.77	8.25
MT-Net	58.72 ± 1.43	32.80 ± 1.35	47.72 ± 1.46	43.11 ± 1.42	45.59	5.5
B-MAML	54.89 ± 1.48	32.53 ± 1.33	53.63 ± 1.37	42.50 ± 1.33	48.39	6
MuMo-MAML	56.82 ± 1.49	33.81 ± 1.36	53.14 ± 1.39	42.22 ± 1.40	46.50	4.75
HSML	55.99 ± 1.41	32.51 ± 1.35	51.26 ± 1.35	42.86 ± 1.42	45.66	6.5
ARML	59.43 ± 1.46	33.30 ± 1.30	56.20 ± 1.34	**45.85 ± 1.46**	48.70	2.25
Ours [1,2,1]	62.37 ± 1.47	**35.23 ± 1.37**	56.20 ± 1.38	44.83 ± 1.43	49.66	2.50
Ours [1,2,2,1]	**64.47 ± 1.41**	33.13 ± 1.37	**58.57 ± 1.39**	45.48 ± 1.30	**50.41**	**2**
Number of shots = 5						
Meta-SGD	67.87 ± 0.74	45.49 ± 0.68	66.84 ± 0.70	52.51 ± 0.81	58.18	8
MAML	68.52 ± 0.79	44.56 ± 0.68	66.18 ± 0.71	42.12 ± 1.36	57.77	7
MT-Net	69.22 ± 0.75	46.57 ± 0.70	63.03 ± 0.69	53.49 ± 0.83	58.08	6
B-MAML	69.01 ± 0.74	46.06 ± 0.69	65.74 ± 0.67	52.43 ± 0.84	58.31	6.75
MuMo-MAML	70.49 ± 0.76	45.89 ± 0.69	67.31 ± 0.68	53.96 ± 0.82	59.41	4.75
HSML	72.07 ± 0.71	44.71 ± 0.66	64.73 ± 0.69	53.38 ± 0.79	58.65	6.75
ARML	71.97 ± 0.70	47.18 ± 0.71	**73.63 ± 0.64**	55.23 ± 0.81	62.00	2.75
Ours [1,2,1]	**74.09 ± 0.73**	47.97 ± 0.68	73.05 ±0.66	56.37 ± 0.81	**62.87**	**1.75**
Ours [1,2,2,1]	72.57 ± 0.74	**48.21 ± 0.67**	72.50 ± 0.65	**56.39 ± 0.80**	62.41	**1.75**

generalization capability of the base learner to novel, unseen datasets. For training the proposed algorithm as well as the baseline methods, we sampled the few-shot tasks τ_i from each of the four datasets, wherein the training set D_i^{tr} and the test set D_i^{ts} contain K classes with N_k^{tr} shots per class.

7.5.2 Setup

We follow the standard protocol for training and evaluating few-shot image classification approaches (Yao et al. 2019). Similar to (Finn, Abbeel, and Levine 2017), we

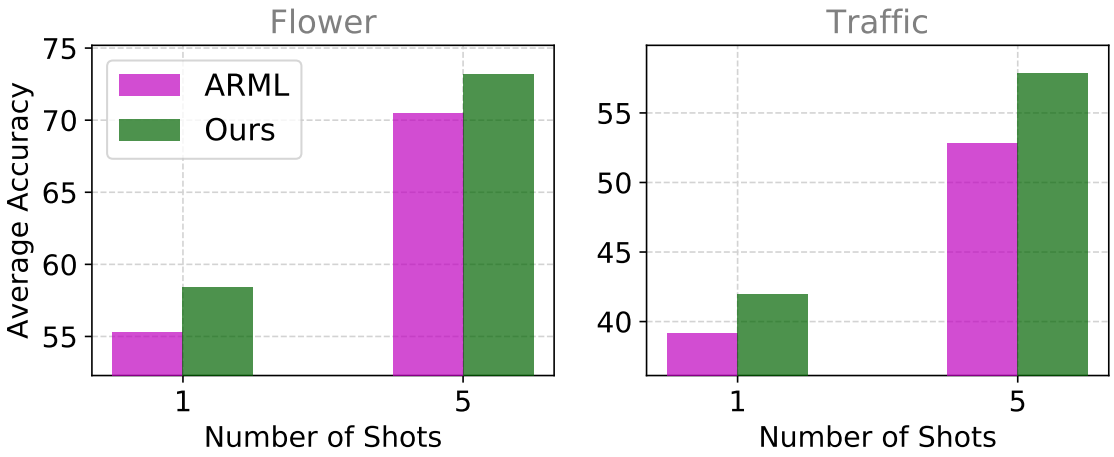


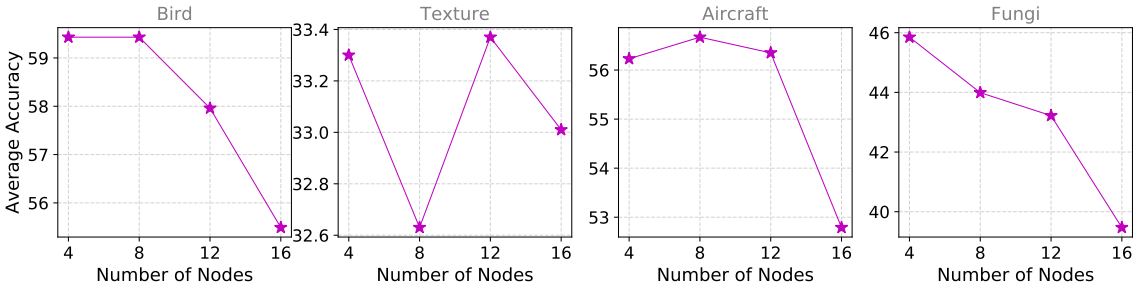
Figure 37. Knowledge graph hierarchies from SGML provide improved inductive biases for generalizing the base learner to even novel unseen datasets.

utilize a simple 4 layer convolutional network as the base learner. For our embedding function $\mathcal{B}$, we use a LeNet (LeCun et al. 1998) model, a popular choice for image feature extraction. The aggregating RNN encoder and decoder functions are constructed using GRU (Chung et al. 2014). Finally, we use a single layer GCN (Kipf and Welling 2017) to implement the "message passing" function between a meta-knowledge graph and a prototype graph. As described in the previous section, all vertices and edges of each knowledge graph in the hierarchy are randomly initialized. Note that, for fair comparison, we regenerated the results for both HSML[6] and ARML[7] at our end using the publicly released codes with the same experiment setup as that of SGML.

[6]https://github.com/huaxiuyao/HSML

[7]https://github.com/huaxiuyao/ARML

[Impact of increasing the number of knowledge graph nodes on ARML (Yao et al. 2020).]



[Impact of choosing different configurations of the knowledge graph hierarchies on SGML.]

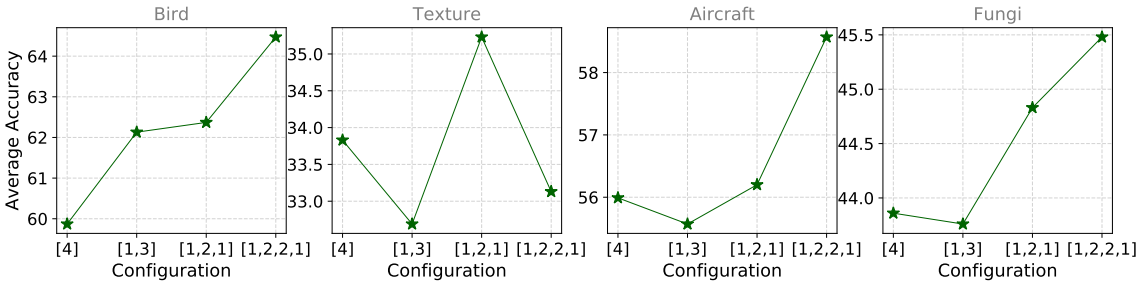


Figure 38. Compared to existing structure-aware meta-learning algorithms, SGML is more flexible to allow richer configurations of knowledge structure. While the performance of ARML declines with increasing number of nodes in the knowledge graph, our approach provides higher performance gains with more complex hierarchies.

7.5.3 Findings

To illustrate the efficacy of representing historical task information in the form of knowledge graph hierarchies, we conduct extensive experiments with different knowledge structures for classification tasks. In particular, we extensively evaluate our proposed method against HSML Yao et al. 2019 and ARML Yao et al. 2020, which are two state-of-the-art methods for using knowledge representation in meta-learning models. To highlight the benefit of structured knowledge representations further, we also compare against several meta-learning methods that do not use have such a component, including Meta-SGD (Z. Li et al. 2017), MAML (Finn, Abbeel,

and Levine 2017), MT-Net (Lee and Choi 2018), B-MAML (Yoon et al. 2018), and MuMo-MAML (Vuorio et al. 2019).

From the results in Table 21 for both 1-shot and 5-shot experiments, we clearly see that the proposed knowledge-graph hierarchies lead to consistently superior performance over existing baselines on average across all datasets. While all structure-aware methods provide non-trivial gains over meta-learning methods that do not perform task-aware modulation, ARML was known to be the best-performing on this benchmark. In comparison, we find that, with the [1,2,2,1] configuration, SGML provides performance gains as high as $\sim 5\%$ on the Bird benchmark over ARML and a boost of $\sim 2\%$ on average across all 4 datasets. We attribute this performance boost to the use of more expressive knowledge graph hierarchies for extracting cross-task relationships. As expected, SGML provides much larger improvements over HSML, $\sim 5\%$ on average with about 8.5% boost on the Bird dataset. We make similar observations for the 5-shot case, where SGML consistently outperforms both HSML and ARML.

The benefits of using a richer knowledge structure is more apparent when the trained base learner is utilized for novel, unseen datasets – *Flower* and *Traffic*. From Figure 37, we observe that our proposed knowledge graph hierarchies provide better inductive biases, as reflected by the significant improvements in the generalization performance when compared to ARML. For a rigorous comparison, in Table 21 we show the results for several other meta-learning baselines and include the *Average Rank* metric (based on relative ranking of the methods for each dataset) for a more holistic understanding of the different approaches.

An important challenge with any structure-aware learning algorithm is determining the appropriate complexity for the knowledge structure. For example, as shown in Figure 38(a), arbitrarily increasing the number of nodes in ARML generally results

in a steady decline in the classification performance. In contrast, knowledge graph hierarchies enable more flexible customization of the structure by either increasing the depth (number of levels L) or changing the number of graphs in each level. Interestingly, from Figure 38, we find that using configurations with larger depth leads to consistent improvements with heterogeneous task distributions.

7.6 Conclusions

In this work, we presented SGML, a novel structure-aware meta-learning algorithm for few-shot classification with heterogeneous tasks. In particular, we introduced the notion of knowledge graph hierarchies to automatically extract cross-task relationships at different levels of complexity for performing effective task-aware modulation. Given a new task with few data samples, our SGML network taps into the knowledge structure to obtain relevant historical task information and tailors meta-initialization parameters using a modulation function.

With extensive empirical studies on the Meta-Dataset benchmark, we showed that the proposed knowledge structure leads to significant performance gains over existing structure-aware meta-learning algorithms. More importantly, SGML also provides improved inductive biases for generalization to unseen datasets and more expressive knowledge representations when compared to state-of-the-art approaches. This work clearly demonstrates the utility of investigating better priors for the knowledge structure to further advance the performance of few-shot learning methods.

Chapter 8

CONCLUSION

In conclusion, this dissertation has rigorously addressed the multifaceted challenges associated with distribution shifts in machine learning, presenting a comprehensive array of innovative methodologies that significantly enhance the adaptability and robustness of AI systems. Through the strategic use of Generative Adversarial Networks (GANs), Vision-Language Models (VLMs), advanced failure estimation mechanisms, and dynamic knowledge graphs, the research has effectively demonstrated robust strategies for overcoming the limitations typically encountered when deploying machine learning models across varied and dynamically evolving data landscapes.

Starting with the innovations presented in Chapter 2, the dissertation introduced SiSTA, an advanced method that leverages generative augmentations for test-time adaptation in scenarios where only minimal target data is available. By fine-tuning StyleGANs and employing novel sampling strategies, this approach efficiently curates synthetic target datasets that closely mirror the characteristics of any target domain, facilitating effective multi-class classification. This breakthrough not only extends the utility of GANs but also sets the stage for future investigations into the behavior of different pruning techniques and the potential expansion of this approach beyond mere classifier adaptation.

Chapter 3 detailed the development of SPHInX, a new approach for addressing ill-posed inverse problems using pre-trained StyleGANv2. This method involves the integration of carefully designed projection heads for style and content latent spaces, coupled with a novel training strategy that ensures accurate and robust embeddings

for even arbitrary OOD images. With extensive empirical studies across multiple datasets, SPHInX has shown significant performance improvements in tasks such as high-resolution image embedding, denoising, and compressed sensing. The findings from this chapter underscore the potential of StyleGAN as a formidable image prior, even in domains where the collection of large-scale datasets for training custom generative models is impractical.

In Chapter 4, the dissertation explored the untapped capabilities of vision-language models, particularly how CLIP can be utilized to revolutionize visual relationship prediction. The CREPE model, which leverages text-based representations and a unique contrastive training strategy, achieved state-of-the-art performance while effectively tackling the long-tail issue prevalent in predicate occurrence distributions. The implications of this work are vast, offering potential advancements in numerous applications such as autonomous navigation and intelligent surveillance systems. However, it also brought to light the ethical considerations necessary when deploying such powerful technologies, particularly in scenarios that could lead to invasive surveillance or biased decision-making.

Chapter 5 introduced PRIME, a novel approach leveraging the foundational strengths of VLMs like CLIP to enhance the detection of failures in pre-trained image classification models. By training an improved version of the classifier (PIM) that learns robust associations between visual features and class-level attributes by projecting into the shared embedding space of VLMs, PRIME can not only identify potential failures but also provide human-interpretable explanations. Extensive experiments across multiple benchmarks have evidenced PRIME's consistent superiority over traditional baselines, showcasing its ability to achieve substantially higher overall scores and better manage trade-offs between failure and success recalls.

Lastly, Chapters 6 and 7 delved deeper into the realm of few-shot learning, presenting CAML and SGML as pioneering methods that utilize dynamic knowledge graphs and structured meta-learning strategies to dramatically enhance model performance in tasks characterized by limited data. By extracting and leveraging cross-task relationships and tailoring meta-initialization parameters, these approaches not only consistently outperformed existing methods but also opened new pathways for further research in enhancing few-shot learning capabilities.

Overall, this body of work not only addresses the immediate challenges posed by distribution shifts but also sets a robust foundation for future research in AI. By bridging gaps between structural data discrepancies and semantic understanding, and by enhancing model reliability through advanced methodologies, this dissertation contributes significantly to the field of machine learning. It promises to influence a wide range of applications and encourages the pursuit of further innovations in this rapidly evolving field, ensuring that AI systems remain resilient, adaptable, and effective even under the most challenging conditions.

8.1 Future Work

For future research building on the foundation laid by this dissertation, several promising avenues can be explored to further enhance the adaptability and functionality of AI systems, particularly in handling out-of-distribution (OOD) data and integrating cross-modal data for richer and more accurate machine learning applications. Here are elaborated future work directions along with additional suggestions:

1. **Network Modulation with GAN Adaptation for Few-Shot Data**: Combining network modulation techniques with GAN adaptation could dramatically

improve the flexibility of GAN networks by dynamically adjusting parameters based on the specific requirements of few-shot data inputs. This approach would be particularly valuable for handling extreme OOD samples, enabling the GAN to adapt its generation process in real-time to produce more accurate representations of sparse or novel data.

2. **Extending SPHInX for Robust Latent Prior Updates**: Building upon SPHInX, extending the method of updating the latent prior to develop a robust sampler capable of generating OOD images without updating the entire GAN structure represents a significant advancement. This approach would allow for continuous adaptation of the latent space, enabling it to more effectively capture and represent diverse and complex data variations.
3. **Enhancing VLMs for Visual Relationship Prediction and Text-to-Image Synthesis**: There is substantial potential to refine text-to-image generation by improving the contextualization of input text, ensuring that it more accurately influences the resulting image generation. This could involve exploring new methods for interpolating on a shared VLM space, allowing for the synthesis of smoothly transitioning images from textual prompts, thereby expanding the creative and practical uses of this technology.
4. **Simultaneous OOD Image Inversion and Interpolation**: Advancing the capability for simultaneous OOD image inversion and interpolation could facilitate the creation of a continuum of image transitions, controlled through text-based prompts interpreted by VLMs. This would enhance the model's ability to handle complex visual manipulations based on nuanced textual input.
5. **Utilizing Knowledge Graphs with LLMs and VLMs for Enhanced Few-Shot Learning**: Expanding the use of knowledge graphs to query and

integrate information from LLMs and VLMs could significantly improve the resolution of few-shot learning tasks. This integration would allow for more precise and contextually appropriate adaptations to new tasks with limited data availability.

6. **Merging Priors from Different Modalities to Create Data-Efficient Multimodal Networks**: Inspired by the success of PRIME, which projects the task model into a VLM subspace, a promising area of future work involves merging priors from different modalities or combining priors with varying strengths. This approach could lead to the creation of multimodal networks that are more data-efficient and capable of leveraging complementary information from diverse data sources. Such networks would be particularly adept at tasks that require a holistic understanding of multiple data types, enhancing the system's overall performance and efficiency.
7. **Ethical AI Deployment and Bias Mitigation**: Ensuring the ethical deployment of AI technologies remains a critical consideration. Future work should continue to address the detection and mitigation of biases in AI models, particularly those that might arise from imbalanced data or skewed interpretations by VLMs, to ensure fair and transparent operation.

By pursuing these directions, future research can not only address the existing challenges but also unlock new potentials for AI systems to operate with enhanced intelligence, adaptability, and ethical integrity in a wide range of real-world applications.

REFERENCES

Abdal, Rameen, Yipeng Qin, and Peter Wonka. 2019. "Image2stylegan: How to embed images into the stylegan latent space?" In *Proceedings of the IEEE/CVF International Conference on Computer Vision,* 4432–4441.

———. 2020. "Image2stylegan++: How to edit the embedded images?" In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition,* 8296–8305.

Abdal, Rameen, Peihao Zhu, Niloy J Mitra, and Peter Wonka. 2021. "Styleflow: Attribute-conditioned exploration of stylegan-generated images using conditional continuous normalizing flows." *ACM Transactions on Graphics (TOG)* 40 (3): 1–21.

Anirudh, Rushil, Jayaraman J Thiagarajan, Bhavya Kailkhura, and Timo Bremer. 2019. "Mimicgan: Robust projection onto image manifolds with corruption mimicking." *arXiv preprint arXiv:1912.07748.*

Bora, Ashish, Ajil Jalal, Eric Price, and Alexandros G Dimakis. 2017. "Compressed sensing using generative models." In *International Conference on Machine Learning,* 537–546. PMLR.

Brock, Andrew, Jeff Donahue, and Karen Simonyan. 2019. "Large Scale GAN Training for High Fidelity Natural Image Synthesis." In *International Conference on Learning Representations.* https://openreview.net/forum?id=B1xsqj09Fm.

Brown, Tom, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. "Language models are few-shot learners." *Advances in neural information processing systems* 33:1877–1901.

Chen, Jiefeng, Frederick Liu, Besim Avci, Xi Wu, Yingyu Liang, and Somesh Jha. 2021. "Detecting errors and estimating accuracy on unlabeled data with self-training ensembles." *Advances in Neural Information Processing Systems* 34:14980–14992.

Chen, Jiefeng, Xi Wu, Yingyu Liang, Somesh Jha, et al. 2020. "Robust Out-of-distribution Detection in Neural Networks." *arXiv preprint arXiv:2003.09711.*

Chen, Tianshui, Weihao Yu, Riquan Chen, and Liang Lin. 2019. "Knowledge-embedded routing network for scene graph generation." In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition,* 6163–6171.

Chen, Ting, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. "A simple framework for contrastive learning of visual representations." In *International conference on machine learning,* 1597–1607. PMLR.

Chib, Pranav Singh, and Pravendra Singh. 2023. "Recent advancements in end-to-end autonomous driving using deep learning: A survey." *IEEE Transactions on Intelligent Vehicles.*

Choi, Yunjey, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. 2020. "StarGAN v2: Diverse Image Synthesis for Multiple Domains." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.*

Chong, Min Jin, and David Forsyth. 2021. "Jojogan: One shot face stylization." *arXiv preprint arXiv:2112.11641.*

Chung, Junyoung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. "Empirical evaluation of gated recurrent neural networks on sequence modeling." *arXiv preprint arXiv:1412.3555.*

Cubuk, Ekin D., Barret Zoph, Dandelion Mané, Vijay Vasudevan, and Quoc V. Le. 2019. "AutoAugment: Learning Augmentation Strategies From Data." In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR),* 113–123. https://doi.org/10.1109/CVPR.2019.00020.

Cukierski, Will. 2013. *Dogs vs. Cats.* https://kaggle.com/competitions/dogs-vs-cats.

Daras, Giannis, Joseph Dean, Ajil Jalal, and Alexandros G Dimakis. 2021. "Intermediate layer optimization for inverse problems using deep generative models." *arXiv preprint arXiv:2102.07364.*

Deng, Ailin, Miao Xiong, and Bryan Hooi. 2023. "Great Models Think Alike: Improving Model Reliability via Inter-Model Latent Agreement." In *Proceedings of the 40th International Conference on Machine Learning,* 202:7675–7693. Proceedings of Machine Learning Research. PMLR, July.

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding." In *NAACL-HLT (1),* 4171–4186. https://aclweb.org/anthology/papers/N/N19/N19-1423/.

DeVries, Terrance, and Graham W Taylor. 2017. "Improved regularization of convolutional neural networks with cutout." *arXiv preprint arXiv:1708.04552.*

Esmaeilpour, Sepideh, Bing Liu, Eric Robertson, and Lei Shu. 2022. "Zero-shot out-of-distribution detection based on the pre-trained model clip." In *Proceedings of the AAAI conference on artificial intelligence,* 36:6568–6576. 6.

Fei, Geli, and B. Liu. 2016. "Breaking the Closed World Assumption in Text Classification." In *NAACL.*

Finn, Chelsea, Pieter Abbeel, and Sergey Levine. 2017. "Model-agnostic meta-learning for fast adaptation of deep networks." In *ICML,* 1126–1135. PMLR.

Finn, Chelsea, and Sergey Levine. 2017. "Meta-learning and universality: Deep representations and gradient descent can approximate any learning algorithm." *arXiv preprint arXiv:1710.11622.*

Finn, Chelsea, Kelvin Xu, and Sergey Levine. 2018. "Probabilistic model-agnostic meta-learning." *Advances in neural information processing systems* 31.

Gal, Yarin, and Zoubin Ghahramani. 2016. "Dropout as a bayesian approximation: Representing model uncertainty in deep learning." In *international conference on machine learning,* 1050–1059. PMLR.

Garg, Saurabh, Sivaraman Balakrishnan, Zachary Chase Lipton, Behnam Neyshabur, and Hanie Sedghi. 2022. "Leveraging unlabeled data to predict out-of-distribution performance." In *International Conference on Learning Representations.* https://openreview.net/forum?id=o_HsiMPYh_x.

Geirhos, Robert, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann. 2020. "Shortcut learning in deep neural networks." *Nature Machine Intelligence* 2 (11): 665–673.

Gokhale, Tejas, Rushil Anirudh, Jayaraman J Thiagarajan, Bhavya Kailkhura, Chitta Baral, and Yezhou Yang. 2023. "Improving Diversity with Adversarially Learned Transformations for Domain Generalization." In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision,* 434–443.

Goodfellow, Ian, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. "Generative adversarial nets." *Advances in neural information processing systems* 27.

Goyal, Sachin, Ananya Kumar, Sankalp Garg, Zico Kolter, and Aditi Raghunathan. 2023. "Finetune like you pretrain: Improved finetuning of zero-shot vision models." In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition,* 19338–19347.

Guillory, Devin, Vaishaal Shankar, Sayna Ebrahimi, Trevor Darrell, and Ludwig Schmidt. 2021. "Predicting with confidence on unseen distributions." In *Proceedings of the IEEE/CVF international conference on computer vision,* 1134–1144.

Guo, Chuan, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. 2017. "On calibration of modern neural networks." In *Proceedings of the 34th International Conference on Machine Learning-Volume 70,* 1321–1330. JMLR. org.

Guo, Jiaxian, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Boyang Li, Dacheng Tao, and Steven Hoi. 2023. "From Images to Textual Prompts: Zero-shot Visual Question Answering with Frozen Large Language Models." In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition,* 10867–10877.

Härkönen, Erik, Aaron Hertzmann, Jaakko Lehtinen, and Sylvain Paris. 2020a. "Ganspace: Discovering interpretable gan controls." *arXiv preprint arXiv:2004.02546.*

———. 2020b. "GANSpace: Discovering Interpretable GAN Controls." In *Advances in Neural Information Processing Systems,* edited by H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, 33:9841–9850. Curran Associates, Inc.

He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016a. "Deep residual learning for image recognition." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,* 770–778.

———. 2016b. "Deep Residual Learning for Image Recognition." In *CVPR.* June.

Hendrycks, Dan, et al. 2021. "The many faces of robustness: A critical analysis of out-of-distribution generalization." In *Proceedings of the IEEE/CVF International Conference on Computer Vision,* 8340–8349.

Hendrycks, Dan, and Thomas Dietterich. 2019. "Benchmarking Neural Network Robustness to Common Corruptions and Perturbations." *Proceedings of the International Conference on Learning Representations.*

Hendrycks, Dan, and Thomas G. Dietterich. 2019. "Benchmarking Neural Network Robustness to Common Corruptions and Perturbations." In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019.* OpenReview.net. https://openreview.net/forum?id=HJz6tiCqYm.

Hendrycks, Dan, and Kevin Gimpel. 2017. "A Baseline for Detecting Misclassified and Out-of-Distribution Examples in Neural Networks." *Proceedings of International Conference on Learning Representations.*

Hendrycks, Dan, Norman Mu, Ekin D Cubuk, Barret Zoph, Justin Gilmer, and Balaji Lakshminarayanan. 2019. "Augmix: A simple data processing method to improve robustness and uncertainty." *arXiv preprint arXiv:1912.02781.*

———. 2020. "AugMix: A Simple Data Processing Method to Improve Robustness and Uncertainty." *Proceedings of the International Conference on Learning Representations (ICLR).*

Hendrycks, Dan, Andy Zou, Mantas Mazeika, Leonard Tang, Bo Li, Dawn Song, and Jacob Steinhardt. 2022. "PixMix: Dreamlike Pictures Comprehensively Improve Safety Measures." *CVPR.*

Hoffman, Judy, et al. 2018. "Cycada: Cycle-consistent adversarial domain adaptation." In *International conference on machine learning,* 1989–1998. Pmlr.

Hosny, Ahmed, Chintan Parmar, John Quackenbush, Lawrence H Schwartz, and Hugo JWL Aerts. 2018. "Artificial intelligence in radiology." *Nature Reviews Cancer* 18 (8): 500–510.

Huang, Jiaxing, Dayan Guan, Aoran Xiao, and Shijian Lu. 2021. "Model adaptation: Historical contrastive learning for unsupervised domain adaptation without source data." *Advances in Neural Information Processing Systems* 34:3635–3649.

Huang, Xun, and Serge Belongie. 2017. "Arbitrary style transfer in real-time with adaptive instance normalization." In *Proceedings of the IEEE International Conference on Computer Vision,* 1501–1510.

Hung, Zih-Siou, Arun Mallya, and Svetlana Lazebnik. 2020. "Contextual translation embedding for visual relationship detection and scene graph generation." *IEEE transactions on pattern analysis and machine intelligence* 43 (11): 3820–3832.

Ilse, Maximilian, Jakub Tomczak, and Max Welling. 2018. "Attention-based deep multiple instance learning." In *International conference on machine learning,* 2127–2136. PMLR.

Ishii, Masato, and Masashi Sugiyama. 2021. "Source-free domain adaptation via distributional alignment by matching batch normalization statistics." *arXiv preprint arXiv:2101.10842.*

Jahanian, Ali, Lucy Chai, and Phillip Isola. 2019. "On the" steerability" of generative adversarial networks." *arXiv preprint arXiv:1907.07171.*

Jain, Saachi, Hannah Lawrence, Ankur Moitra, and Aleksander Madry. 2023. "Distilling Model Failures as Directions in Latent Space." In *The Eleventh International Conference on Learning Representations.* https://openreview.net/forum?id=99RpBVpLiX.

Jiang, Yiding, Vaishnavh Nagarajan, Christina Baek, and J Zico Kolter. 2022. "Assessing Generalization of SGD via Disagreement." In *International Conference on Learning Representations.* https://openreview.net/forum?id=WvOGCEAQhxl.

Jiang, Yiding, Behnam Neyshabur, Hossein Mobahi, Dilip Krishnan, and Samy Bengio. 2019. "Fantastic generalization measures and where to find them." *arXiv preprint arXiv:1912.02178.*

Joshi, Nitish, Xiang Pan, and He He. 2022. "Are all spurious features in natural language alike? an analysis through a causal lens." *arXiv preprint arXiv:2210.14011.*

Kafri, Omer, Or Patashnik, Yuval Alaluf, and Daniel Cohen-Or. 2021. "Stylefusion: A generative model for disentangling spatial segments." *arXiv preprint arXiv:2107.07437.*

Kang, Kyoungkook, Seongtae Kim, and Sunghyun Cho. 2021. "GAN Inversion for Out-of-Range Images with Geometric Transformations." In *Proceedings of the IEEE/CVF International Conference on Computer Vision,* 13941–13949.

Karras, Tero, Timo Aila, Samuli Laine, and Jaakko Lehtinen. 2017a. "Progressive growing of gans for improved quality, stability, and variation." *arXiv preprint arXiv:1710.10196.*

———. 2017b. "Progressive Growing of GANs for Improved Quality, Stability, and Variation." *CoRR* abs/1710.10196. arXiv: 1710.10196. http://arxiv.org/abs/1710.10196.

Karras, Tero, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. 2021. "Alias-free generative adversarial networks." *arXiv preprint arXiv:2106.12423.*

Karras, Tero, Samuli Laine, and Timo Aila. 2019. "A style-based generator architecture for generative adversarial networks." In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition,* 4401–4410.

Karras, Tero, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. 2020. "Analyzing and improving the image quality of stylegan." In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition,* 8110–8119.

Khattak, Muhammad Uzair, Hanoona Rasheed, Muhammad Maaz, Salman Khan, and Fahad Shahbaz Khan. 2023. "Maple: Multi-modal prompt learning." In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition,* 19113–19122.

Kipf, Thomas N, and Max Welling. 2017. "Semi-supervised classification with graph convolutional networks." In *ICLR.*

Kirsch, Andreas, Jishnu Mukhoti, Joost van Amersfoort, Philip H. S. Torr, and Yarin Gal. 2021. *On Pitfalls in OoD Detection: Entropy Considered Harmful.* Uncertainty and Robustness in Deep Learning Workshop, ICML.

Koch, Gregory, Richard Zemel, Ruslan Salakhutdinov, et al. 2015. "Siamese neural networks for one-shot image recognition." In *ICML deep learning workshop,* vol. 2. Lille.

Krishna, Ranjay, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. 2017. "Visual genome: Connecting language and vision using crowdsourced dense image annotations." *International journal of computer vision* 123:32–73.

Krizhevsky, Alex, Geoffrey Hinton, et al. 2009. *Learning multiple layers of features from tiny images.* Technical Report. Toronto, ON, Canada: University of Toronto.

Kundu, Jogendra Nath, Naveen Venkat, Ambareesh Revanur, R Venkatesh Babu, et al. 2020. "Towards inheritable models for open-set domain adaptation." In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition,* 12376–12385.

LeCun, Yann, Léon Bottou, Yoshua Bengio, and Patrick Haffner. 1998. "Gradient-based learning applied to document recognition." *Proceedings of the IEEE* 86 (11): 2278–2324.

Lee, Yoonho, Annie S Chen, Fahim Tajwar, Ananya Kumar, Huaxiu Yao, Percy Liang, and Chelsea Finn. 2022. "Surgical fine-tuning improves adaptation to distribution shifts." *arXiv preprint arXiv:2210.11466.*

Lee, Yoonho, and Seungjin Choi. 2018. "Gradient-based meta-learning with learned layerwise metric and subspace." In *International Conference on Machine Learning,* 2927–2936. PMLR.

Li, Da, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. 2017. "Deeper, broader and artier domain generalization." In *Proceedings of the IEEE international conference on computer vision,* 5542–5550.

Li, Junnan, Dongxu Li, Caiming Xiong, and Steven Hoi. 2022. "Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation." In *International Conference on Machine Learning,* 12888–12900. PMLR.

Li, Zhenguo, Fengwei Zhou, Fei Chen, and Hang Li. 2017. "Meta-sgd: Learning to learn quickly for few-shot learning." *arXiv preprint arXiv:1707.09835.*

Liang, Jian, Dapeng Hu, and Jiashi Feng. 2020. "Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation." In *International conference on machine learning,* 6028–6039. PMLR.

Liu, Vivian, and Lydia B Chilton. 2022. "Design guidelines for prompt engineering text-to-image generative models." In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems,* 1–23.

Liu, Weitang, Xiaoyun Wang, John Owens, and Yixuan Li. 2020. "Energy-based out-of-distribution detection." *Advances in neural information processing systems* 33:21464–21475.

Liu, Xinyu, and Yixuan Yuan. 2022. "A source-free domain adaptive polyp detection framework with style diversification flow." *IEEE Transactions on Medical Imaging* 41 (7): 1897–1908.

Liu, Ziwei, Ping Luo, Xiaogang Wang, and Xiaoou Tang. 2015. "Deep Learning Face Attributes in the Wild." In *Proceedings of International Conference on Computer Vision (ICCV).* December.

Menon, Sachit, Alexandru Damian, Shijia Hu, Nikhil Ravi, and Cynthia Rudin. 2020. "Pulse: Self-supervised photo upsampling via latent space exploration of generative models." In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition,* 2437–2445.

Merullo, Jack, Louis Castricato, Carsten Eickhoff, and Ellie Pavlick. 2022. "Linearly mapping from image to text space." *arXiv preprint arXiv:2209.15162.*

Michels, Felix, Nikolas Adaloglou, Tim Kaiser, and Markus Kollmann. 2023. "Contrastive Language-Image Pretrained (CLIP) Models are Powerful Out-of-Distribution Detectors." *arXiv preprint arXiv:2303.05828.*

Minderer, Matthias, Josip Djolonga, Rob Romijnders, Frances Hubis, Xiaohua Zhai, Neil Houlsby, Dustin Tran, and Mario Lucic. 2021. "Revisiting the calibration of modern neural networks." *Advances in Neural Information Processing Systems* 34:15682–15694.

Ming, Yifei, Ziyang Cai, Jiuxiang Gu, Yiyou Sun, Wei Li, and Yixuan Li. 2022. "Delving into Out-of-Distribution Detection with Vision-Language Representations." In *Advances in Neural Information Processing Systems,* edited by Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho. https://openreview.net/forum?id=KnCS9390Va.

Mitra, Sinjini, Rakshith Subramanyam, Rushil Anirudh, Jayaraman J Thiagarajan, Ankita Shukla, and Pavan K Turaga. 2023. "Adapting Blackbox Generative Models via Inversion." In *Proceedings of the ICML Workshops.*

Montanari, Andrea, and Basil N Saeed. 2022. "Universality of empirical risk minimization." In *Conference on Learning Theory,* 4310–4312. PMLR.

Munkhdalai, Tsendsuren, and Hong Yu. 2017. "Meta networks." In *ICML,* 2554–2563. PMLR.

Nagabandi, Anusha, Ignasi Clavera, Simin Liu, Ronald S Fearing, Pieter Abbeel, Sergey Levine, and Chelsea Finn. 2018. "Learning to adapt in dynamic, real-world environments through meta-reinforcement learning." *arXiv preprint arXiv:1803.11347.*

Nakano, Reiichiro, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, et al. 2021. "Webgpt: Browser-assisted question-answering with human feedback." *arXiv preprint arXiv:2112.09332.*

Narayanaswamy, Vivek, Rushil Anirudh, Irene Kim, Yamen Mubarka, Andreas Spanias, and Jayaraman J. Thiagarajan. 2022. "Predicting the Generalization Gap in Deep Models using Anchoring." In *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP),* 4393–4397.

Ng, Nathan, Kyunghyun Cho, Neha Hulkund, and Marzyeh Ghassemi. 2022. "Predicting out-of-domain generalization with local manifold smoothness." *arXiv preprint arXiv:2207.02093.*

Ngo, Chi T, Nora S Newcombe, and Ingrid R Olson. 2018. "The ontogeny of relational memory and pattern separation." *Developmental science* 21 (2): e12556.

Oord, Aaron van den, Yazhe Li, and Oriol Vinyals. 2018. "Representation learning with contrastive predictive coding." *arXiv preprint arXiv:1807.03748.*

Patashnik, Or, Zongze Wu, Eli Shechtman, Daniel Cohen-Or, and Dani Lischinski. 2021. "Styleclip: Text-driven manipulation of stylegan imagery." In *Proceedings of the IEEE/CVF International Conference on Computer Vision,* 2085–2094.

Peng, Xingchao, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. 2019a. "Moment matching for multi-source domain adaptation." In *Proceedings of the IEEE/CVF international conference on computer vision,* 1406–1415.

———. 2019b. "Moment matching for multi-source domain adaptation." In *Proceedings of the IEEE/CVF international conference on computer vision,* 1406–1415.

Peyre, Julia, Josef Sivic, Ivan Laptev, and Cordelia Schmid. 2017. "Weakly-supervised learning of visual relations." In *Proceedings of the ieee international conference on computer vision,* 5179–5188.

Plumerault, Antoine, Hervé Le Borgne, and Céline Hudelot. 2020. "Controlling generative models with continuous factors of variations." *arXiv preprint arXiv:2001.10238.*

Pratt, Sarah, Ian Covert, Rosanne Liu, and Ali Farhadi. 2023. "What does a platypus look like? generating customized prompts for zero-shot image classification." In *Proceedings of the IEEE/CVF International Conference on Computer Vision,* 15691–15701.

Qu, Haoxuan, Yanchao Li, Lin Geng Foo, Jason Kuen, Jiuxiang Gu, and Jun Liu. 2022. "Improving the reliability for confidence estimation." In *European Conference on Computer Vision,* 391–408. Springer.

Radford, Alec, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021a. "Learning transferable visual models from natural language supervision." In *International conference on machine learning,* 8748–8763. PMLR.

———. 2021b. "Learning transferable visual models from natural language supervision." In *International conference on machine learning,* 8748–8763. PMLR.

Radford, Alec, Luke Metz, and Soumith Chintala. 2016. "Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks." In *4th International Conference on Learning Representations.*

Radford, Alec, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. "Language models are unsupervised multitask learners." *OpenAI blog* 1 (8): 9.

Rahman, Aimon, M. Sohel Rahman, and Mahdy Rahman Chowdhury Mahdy. 2021. "3C-GAN: class-consistent CycleGAN for malaria domain adaptation model." *Biomedical Physics & Engineering Express* 7.

Raj, Ankit, Yuqi Li, and Yoram Bresler. 2019. "Gan-based projector for faster recovery with convergence guarantees in linear inverse problems." In *Proceedings of the IEEE/CVF International Conference on Computer Vision,* 5602–5611.

Rao, Qing, and Jelena Frtunikj. 2018. "Deep Learning for Self-Driving Cars: Chances and Challenges." In *Proceedings of the 1st International Workshop on Software Engineering for AI in Autonomous Systems,* 35–38. SEFAIS '18. Gothenburg, Sweden: Association for Computing Machinery. https://doi.org/10.1145/3194085.3194087.

Ren, Shaoqing, Kaiming He, Ross Girshick, and Jian Sun. 2015. "Faster r-cnn: Towards real-time object detection with region proposal networks." *Advances in neural information processing systems* 28.

Richardson, Elad, Yuval Alaluf, Or Patashnik, Yotam Nitzan, Yaniv Azar, Stav Shapiro, and Daniel Cohen-Or. 2021. "Encoding in style: a stylegan encoder for image-to-image translation." In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition,* 2287–2296.

Robey, Alexander, George J Pappas, and Hamed Hassani. 2021. "Model-based domain generalization." *Advances in Neural Information Processing Systems* 34:20210–20229.

Rombach, Robin, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022. "High-resolution image synthesis with latent diffusion models." In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition,* 10684–10695.

Russakovsky, Olga, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, et al. 2015. "ImageNet Large Scale Visual Recognition

Challenge." *International Journal of Computer Vision (IJCV)* 115 (3): 211–252. https://doi.org/10.1007/s11263-015-0816-y.

Saharia, Chitwan, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S Sara Mahdavi, Rapha Gontijo Lopes, et al. 2022. "Photorealistic text-to-image diffusion models with deep language understanding." *arXiv preprint arXiv:2205.11487.*

Sankaranarayanan, Swami, Yogesh Balaji, Carlos D Castillo, and Rama Chellappa. 2018. "Generate to adapt: Aligning domains using generative adversarial networks." In *Proceedings of the IEEE conference on computer vision and pattern recognition,* 8503–8512.

Santoro, Adam, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy Lillicrap. 2016. "Meta-learning with memory-augmented neural networks." In *International conference on machine learning,* 1842–1850. PMLR.

Sauer, Axel, Katja Schwarz, and Andreas Geiger. 2022. "Stylegan-xl: Scaling stylegan to large diverse datasets." In *ACM SIGGRAPH 2022 conference proceedings,* 1–10.

Schwenk, Dustin, Apoorv Khandelwal, Christopher Clark, Kenneth Marino, and Roozbeh Mottaghi. 2022. "A-okvqa: A benchmark for visual question answering using world knowledge." In *European Conference on Computer Vision,* 146–162. Springer.

Shah, Viraj, and Chinmay Hegde. 2018. "Solving linear inverse problems using gan priors: An algorithm with provable guarantees." In *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP),* 4609–4613. IEEE.

Simonyan, Karen, and Andrew Zisserman. 2015. "Very Deep Convolutional Networks for Large-Scale Image Recognition." In *Proceedings of the International Conference on Learning Representations.*

Snell, Jake, Kevin Swersky, and Richard S Zemel. 2017. "Prototypical networks for few-shot learning." In *NeurIPS.*

Song, Guoxian, Linjie Luo, Jing Liu, Wan-Chun Ma, Chunpong Lai, Chuanxia Zheng, and Tat-Jen Cham. 2021. "AgileGAN: Stylizing Portraits by Inversion-Consistent Transfer Learning." *ACM Transactions on Graphics (Proc. SIGGRAPH)* (July).

Song, Haoyu, Li Dong, Wei-Nan Zhang, Ting Liu, and Furu Wei. 2022. "Clip models are few-shot learners: Empirical studies on vqa and visual entailment." *arXiv preprint arXiv:2203.07190.*

Steiner, Andreas, Alexander Kolesnikov, Xiaohua Zhai, Ross Wightman, Jakob Uszkoreit, and Lucas Beyer. 2021. "How to train your vit? data, augmentation, and regularization in vision transformers." *arXiv preprint arXiv:2106.10270.*

Subramanyam, R, M Heimann, TS Jayram, R Anirudh, B Kailkhura, M Naufel, and JJ Thiagarajan. 2021. *Learning Knowledge Graph Hierarchies for Improving Few-Shot Classification.* Technical report. Lawrence Livermore National Lab.(LLNL), Livermore, CA (United States).

Subramanyam, Rakshith. 2018. "Chartopolis: A Self Driving Car Test Bed." Master's thesis, Department of Electrical Engineering, Arizona State University.

Subramanyam, Rakshith, Mark Heimann, TS Jayram, Rushil Anirudh, and Jayaraman J Thiagarajan. 2023. "Contrastive knowledge-augmented meta-learning for few-shot classification." In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision,* 2479–2487.

Subramanyam, Rakshith, TS Jayram, Rushil Anirudh, and Jayaraman J Thiagarajan. 2023. "CREPE: Learnable Prompting With CLIP Improves Visual Relationship Prediction." *arXiv preprint arXiv:2307.04838.*

Subramanyam, Rakshith, Vivek Narayanaswamy, Mark Naufel, Andreas Spanias, and Jayaraman J Thiagarajan. 2022. "Improved StyleGAN-v2 based Inversion for Out-of-Distribution Images." In *International Conference on Machine Learning,* 20625–20639. PMLR.

Subramanyam, Rakshith, Kowshik Thopalli, Spring Berman, Pavan Turaga, and Jayaraman J Thiagarajan. 2023. "Single-Shot Domain Adaptation via Target-Aware Generative Augmentations." In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP),* 1–5. IEEE.

Sun, Yu, Xiaolong Wang, Zhuang Liu, John Miller, Alexei Efros, and Moritz Hardt. 2020. "Test-time training with self-supervision for generalization under distribution shifts." In *International conference on machine learning,* 9229–9248. PMLR.

Tang, Kaihua, Yulei Niu, Jianqiang Huang, Jiaxin Shi, and Hanwang Zhang. 2020. "Unbiased scene graph generation from biased training." In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition,* 3716–3725.

Tang, Kaihua, Hanwang Zhang, Baoyuan Wu, Wenhan Luo, and Wei Liu. 2019. "Learning to compose dynamic tree structures for visual contexts." In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition,* 6619–6628.

Tang, Song, Yan Yang, Zhiyuan Ma, Norman Hendrich, Fanyu Zeng, Shuzhi Sam Ge, Changshui Zhang, and Jianwei Zhang. 2021. "Nearest neighborhood-based deep clustering for source data-absent unsupervised domain adaptation." *arXiv preprint arXiv:2107.12585.*

Tewari, Ayush, Mohamed Elgharib, Florian Bernard, Hans-Peter Seidel, Patrick Pérez, Michael Zollhöfer, and Christian Theobalt. 2020. "Pie: Portrait image embedding for semantic control." *ACM Transactions on Graphics (TOG)* 39 (6): 1–14.

Tewari, Ayush, Mohamed Elgharib, Gaurav Bharaj, Florian Bernard, Hans-Peter Seidel, Patrick Pérez, Michael Zollhofer, and Christian Theobalt. 2020. "Stylerig: Rigging stylegan for 3d control over portrait images." In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition,* 6142–6151.

Thopalli, Kowshik, Rakshith Subramanyam, Pavan Turaga, and Jayaraman J Thiagarajan. 2023. "Target-Aware Generative Augmentations for Single-Shot Adaptation." *arXiv preprint arXiv:2305.13284.*

Thopalli, Kowshik, Pavan Turaga, and Jayaraman J Thiagarajan. 2022. "Domain Alignment Meets Fully Test-Time Adaptation." In *Asian Conference on Machine Learning, 2022.*

Thrun, Sebastian, and Lorien Pratt. 2012. *Learning to learn.* Springer Science & Business Media.

Torralba, Antonio, and Alexei A Efros. 2011. "Unbiased look at dataset bias." In *CVPR 2011,* 1521–1528. IEEE.

Touvron, Hugo, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. "Llama 2: Open foundation and fine-tuned chat models." *arXiv preprint arXiv:2307.09288.*

Tov, Omer, Yuval Alaluf, Yotam Nitzan, Or Patashnik, and Daniel Cohen-Or. 2021. "Designing an encoder for stylegan image manipulation." *ACM Transactions on Graphics (TOG)* 40 (4): 1–14.

Triantafillou, Eleni, Hugo Larochelle, Richard Zemel, and Vincent Dumoulin. 2021. "Learning a universal template for few-shot dataset generalization." In *International Conference on Machine Learning,* 10424–10433. PMLR.

Triantafillou, Eleni, Tyler Zhu, Vincent Dumoulin, Pascal Lamblin, Utku Evci, Kelvin Xu, Ross Goroshin, Carles Gelada, Kevin Swersky, Pierre-Antoine Manzagol, et al. 2020. "Meta-dataset: A dataset of datasets for learning to learn from few examples." In *ICLR.*

Trivedi, Puja, Danai Koutra, and Jayaraman J Thiagarajan. 2023. "A Closer Look At Scoring Functions And Generalization Prediction." In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP),* 1–5. IEEE.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. "Attention is all you need." In *Advances in Neural Information Processing Systems,* 6000–6010.

Vinyals, Oriol, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. 2016. "Matching networks for one shot learning." In *NeurIPS,* 29:3630–3638.

Voynov, Andrey, and Artem Babenko. 2020. "Unsupervised discovery of interpretable directions in the gan latent space." In *International Conference on Machine Learning,* 9786–9796. PMLR.

Vuorio, Risto, Shao-Hua Sun, Hexiang Hu, and Joseph J Lim. 2019. "Multimodal model-agnostic meta-learning via task-aware modulation." *arXiv preprint arXiv:1910.13616.*

Wang, Dequan, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. 2021. "Tent: Fully Test-Time Adaptation by Entropy Minimization." In *International Conference on Learning Representations.* https://openreview.net/forum?id=uXl3bZLkr3c.

Wang, Hualiang, Yi Li, Huifeng Yao, and Xiaomeng Li. 2023. "Clipn for zero-shot ood detection: Teaching clip to say no." In *Proceedings of the IEEE/CVF International Conference on Computer Vision,* 1802–1812.

Wang, Tengfei, Yong Zhang, Yanbo Fan, Jue Wang, and Qifeng Chen. 2021. "High-Fidelity GAN Inversion for Image Attribute Editing." *arXiv preprint arXiv:2109.06590.*

Wang, Yaxing, Abel Gonzalez-Garcia, David Berga, Luis Herranz, Fahad Shahbaz Khan, and Joost van de Weijer. 2020. "Minegan: effective knowledge transfer from gans to target domains with few images." In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition,* 9332–9341.

Wei, Jason, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V Le. 2022. "Finetuned Language Models are Zero-Shot Learners." In *International Conference on Learning Representations.* https://openreview.net/forum?id=gEZrGCozdqR.

Wei, Yixuan, Han Hu, Zhenda Xie, Ze Liu, Zheng Zhang, Yue Cao, Jianmin Bao, Dong Chen, and Baining Guo. 2023. "Improving CLIP Fine-tuning Performance." In *Proceedings of the IEEE/CVF International Conference on Computer Vision,* 5439–5449.

Wing, Erik A, Maria C D'Angelo, Asaf Gilboa, and Jennifer D Ryan. 2021. "The Role of the Ventromedial Prefrontal Cortex and Basal Forebrain in Relational Memory and Inference." *Journal of Cognitive Neuroscience,* 1–14.

Wortsman, Mitchell, Gabriel Ilharco, Jong Wook Kim, Mike Li, Simon Kornblith, Rebecca Roelofs, Raphael Gontijo Lopes, Hannaneh Hajishirzi, Ali Farhadi, Hongseok Namkoong, et al. 2022. "Robust fine-tuning of zero-shot models." In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition,* 7959–7971.

Wu, Zongze, Dani Lischinski, and Eli Shechtman. 2021. "Stylespace analysis: Disentangled controls for stylegan image generation." In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition,* 12863–12872.

Wu, Zongze, Yotam Nitzan, Eli Shechtman, and Dani Lischinski. 2021. "Stylealign: Analysis and applications of aligned stylegan models." *arXiv preprint arXiv:2110.11323.*

Wulff, Jonas, and Antonio Torralba. 2020. "Improving inversion and generation diversity in stylegan using a gaussianized latent space." *arXiv preprint arXiv:2009.06529.*

Xia, Weihao, Yulun Zhang, Yujiu Yang, Jing-Hao Xue, Bolei Zhou, and Ming-Hsuan Yang. 2022. "Gan inversion: A survey." *IEEE Transactions on Pattern Analysis and Machine Intelligence.*

Xu, Danfei, Yuke Zhu, Christopher B Choy, and Li Fei-Fei. 2017. "Scene graph generation by iterative message passing." In *Proceedings of the IEEE conference on computer vision and pattern recognition,* 5410–5419.

Xu, Zhenlin, Deyi Liu, Junlin Yang, Colin Raffel, and Marc Niethammer. 2021. "Robust and Generalizable Visual Representation Learning via Random Convolutions." In *International Conference on Learning Representations.* https://openreview.net/forum?id=BVSM0x3EDK6.

Yang, Jianwei, Jiasen Lu, Stefan Lee, Dhruv Batra, and Devi Parikh. 2018. "Graph r-cnn for scene graph generation." In *Proceedings of the European conference on computer vision (ECCV),* 670–685.

Yang, Shiqi, Joost van de Weijer, Luis Herranz, Shangling Jui, et al. 2021. "Exploiting the intrinsic neighborhood structure for source-free domain adaptation." *Advances in Neural Information Processing Systems* 34:29393–29405.

Yang, Yuzhe, Haoran Zhang, Dina Katabi, and Marzyeh Ghassemi. 2023. "Change is Hard: A Closer Look at Subpopulation Shift." In *International Conference on Machine Learning.*

Yao, Huaxiu, Ying Wei, Junzhou Huang, and Zhenhui Li. 2019. "Hierarchically structured meta-learning." In *International Conference on Machine Learning,* 7045–7054. PMLR.

Yao, Huaxiu, Xian Wu, Zhiqiang Tao, Yaliang Li, Bolin Ding, Ruirui Li, and Zhenhui Li. 2020. "Automated relational meta-learning." In *ICLR.*

Yeh, Raymond A, Chen Chen, Teck Yian Lim, Alexander G Schwing, Mark Hasegawa-Johnson, and Minh N Do. 2017. "Semantic image inpainting with deep generative models." In *Proceedings of the IEEE conference on computer vision and pattern recognition,* 5485–5493.

Yoon, Jaesik, Taesup Kim, Ousmane Dia, Sungwoong Kim, Yoshua Bengio, and Sungjin Ahn. 2018. "Bayesian model-agnostic meta-learning." In *Proceedings of the 32nd International Conference on Neural Information Processing Systems,* 7343–7353.

Young, Albert T, Mulin Xiong, Jacob Pfau, Michael J Keiser, and Maria L Wei. 2020. "Artificial intelligence in dermatology: a primer." *Journal of Investigative Dermatology* 140 (8): 1504–1512.

Yu, Qifan, Juncheng Li, Yu Wu, Siliang Tang, Wei Ji, and Yueting Zhuang. 2023. "Visually-Prompted Language Model for Fine-Grained Scene Graph Generation in an Open World." *arXiv preprint arXiv:2303.13233.*

Yu, Shoubin, Jaemin Cho, Prateek Yadav, and Mohit Bansal. 2024. "Self-chained image-language model for video localization and question answering." *Advances in Neural Information Processing Systems* 36.

Yue, Fei, Chao Zhang, MingYang Yuan, Chen Xu, and YaLin Song. 2022. "Survey of Image Augmentation Based on Generative Adversarial Network." *Journal of Physics: Conference Series* 2203, no. 1 (February): 012052. https://doi.org/10.1088/1742-6596/2203/1/012052.

Yun, Sangdoo, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. 2019a. "Cutmix: Regularization strategy to train strong classifiers with localizable features." In *Proceedings of the IEEE/CVF international conference on computer vision,* 6023–6032.

———. 2019b. "CutMix: Regularization Strategy to Train Strong Classifiers with Localizable Features." In *International Conference on Computer Vision (ICCV).* Published.

Zellers, Rowan, Mark Yatskar, Sam Thomson, and Yejin Choi. 2018. "Neural motifs: Scene graph parsing with global context." In *Proceedings of the IEEE conference on computer vision and pattern recognition,* 5831–5840.

Zhang, Hongyi, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. 2018. "mixup: Beyond Empirical Risk Minimization." In *International Conference on Learning Representations.*

Zhang, Marvin, Sergey Levine, and Chelsea Finn. 2021. "Memo: Test time robustness via adaptation and augmentation." *arXiv preprint arXiv:2110.09506.*

Zhang, Richard, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. 2018. "The unreasonable effectiveness of deep features as a perceptual metric." In *Proceedings of the IEEE conference on computer vision and pattern recognition,* 586–595.

Zhao, Tiancheng, Tianqi Zhang, Mingwei Zhu, Haozhan Shen, Kyusong Lee, Xiaopeng Lu, and Jianwei Yin. 2022. "Vl-checklist: Evaluating pre-trained vision-language models with objects, attributes and relations." *arXiv preprint arXiv:2207.00221.*

Zhou, Kaiyang, Jingkang Yang, Chen Change Loy, and Ziwei Liu. 2022. "Conditional prompt learning for vision-language models." In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition,* 16816–16825.

Zhu, Fei, Zhen Cheng, Xu-Yao Zhang, and Cheng-Lin Liu. 2022. "Rethinking confidence calibration for failure prediction." In *European Conference on Computer Vision,* 518–536. Springer.

Zhu, Jiapeng, Yujun Shen, Deli Zhao, and Bolei Zhou. 2020. "In-domain gan inversion for real image editing." In *European conference on computer vision,* 592–608. Springer.

Zhu, Peihao, Rameen Abdal, Yipeng Qin, John Femiani, and Peter Wonka. 2020. "Improved StyleGAN Embedding: Where are the Good Latents?" *arXiv preprint arXiv:2012.09036.*

Zhu, Yi, Zhaoqing Zhu, Bingqian Lin, Xiaodan Liang, Feng Zhao, and Jianzhuang Liu. 2022. "RelCLIP: Adapting Language-Image Pretraining for Visual Relationship Detection via Relational Contrastive Learning." In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing,* 4800–4810.